

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО”**  
**ТЕПЛОЕНЕРГЕТИЧНИЙ ФАКУЛЬТЕТ**  
Кафедра автоматизації проектування енергетичних процесів і систем

“До захисту допущено”

Завідувач кафедри

\_\_\_\_\_  
(підпис) О.В. Коваль  
(ініціали, прізвище)

“    ”    \_\_\_\_\_ 2019 р.

## **Магістерська дисертація**

зі спеціальності 121 Інженерія програмного забезпечення

за спеціалізацією Інженерія програмного забезпечення розподілених систем

на тему: Інструментальні засоби визначення гідроакустичної сигнатури з обробкою в MATLAB

Виконала: студентка 6 курсу, групи ТВ-82мп  
(шифр групи)

\_\_\_\_\_  
Костенко Олександра Павлівна  
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к.т.н., доцент Кублій Л.І.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ — 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності — 121 Інженерія програмного забезпечення

за спеціалізацією — Програмне забезпечення розподілених систем

ЗАТВЕРДЖУЮ  
Завідувач кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” ” \_\_\_\_\_ 2019р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
Костенко Олександрі Павлівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Інструментальні засоби визначення гідроакустичної сигнатури з обробкою в MATLAB

керівник роботи \_\_\_\_\_ Кублій Лариса Іванівна, к.т.н., доцент  
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від 4 листопада 2019 р. № 3812-с

2. Строк подання студентом роботи \_\_10\_\_ грудня 2019 р.

3. Об'єкт дослідження Алгоритм визначення гідроакустичної сигнатури з обробкою в MATLAB.

4. Предмет дослідження Визначення гідроакустичної сигнатури та аналіз отриманих результатів.

5. Перелік питань, які потрібно розробити: Проаналізувати актуальність обробки гідроакустичних сигналів. Проаналізувати аналогічні програмні системи та їхні недоліки. Реалізувати алгоритми обробки параметрів вхідного сигналу. Завантажувати, оброблювати та записувати результати обчислень та інші дані до бази даних. Розробити користувацький інтерфейс для користувача. Розробити можливість графічного подання отриманих результатів.

6. Орієнтовний перелік ілюстративного матеріалу: вхідні та вихідні дані для обробки гідроакустичного сигналу, архітектура програмного додатку, графічне подання результатів проведених розрахунків.

7. Орієнтований перелік публікацій: Дослідження можливостей взаємодії застосунків C# і MATLAB (“Сталий розвиток — XXI століття: управління, технології, моделі.”)

8. Дата видачі завдання ” \_\_\_\_ ” \_\_\_\_\_ 2019 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. — 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 — 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 — 29.03.19 р.	
6	Розробка програмного продукту	03.06 — 25.10.19 р.	
5	Переддипломна практика	02.09 — 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 — 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист	16.12.19 р.	

Студент \_\_\_\_\_ Костенко О.П.  
(підпис) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Кублій Л.І.  
(підпис) (прізвище та ініціали)

# РЕФЕРАТ

## **Структура і обсяг дипломної роботи**

Магістерська дисертація складається зі вступу, шести розділів і висновку.

Робота містить в собі посилання на 40 джерел, 28 ілюстрацій, 24 таблиці.

Основна частина роботи викладена на 92 сторінках.

## **Актуальність теми**

Актуальність даної роботи пояснюється існуючою проблемою визначення типів об'єктів у товщі води в гідроакустиці.

Дана робота присвячена розробці програмного засобу для розрахунку гідроакустичної сигнатури з обробкою в Matlab.

**Метою дослідження** є створення програмного додатку для визначення гідроакустичної сигнатури.

Для досягнення поставленої мети потрібно виконати такі **задачі дослідження**:

- проаналізувати актуальність використання гідроакустичних сигналів;
- проаналізувати аналогічні програмні системи та їхні недоліки;
- реалізувати математичну обробку гідроакустичної сигнатури;
- налаштувати зв'язок між додатками та записувати результати обчислень до бази даних;
- розробити користувацький інтерфейс;
- розробити засоби графічного подання отриманих результатів;

**Об'єкт дослідження**: принцип обробки гідроакустичного сигналу, підходи та алгоритми розрахунку.

**Предмет дослідження**: обробка гідроакустичного сигналу, аналіз отриманих результатів.

**Методи дослідження**: методи математичного й фізичного моделювання.

**Наукова новизна**: перша комплексна система для розрахунку гідроакустичної сигнатури, що надає можливість зберігати оброблений сигнал і порівнювати кілька оброблених сигналів.

**Практичне значення:** розрахунок гідроакустичної сигнатури дасть можливість користувачам даної системи аналізувати товщу моря чи океану, виконувати аналіз для визначення сусіднього об'єкта.

**Апробація:** результати досліджень стосовно системи загалом представлені на V Міжнародній науково-практичній конференції “Сталий розвиток XXI століття: управління, технології, моделі (наукові читання імені Ігоря Недіна)” 2018 року.

**Ключові слова:** ОБРОБКА СИГНАЛУ, MATLAB, C#, ВЗАЄМОДІЯ ЗАСТОСУНКІВ.

## ABSTRACT

### **Structure and volume of the thesis**

The master's thesis consists of an introduction, six sections and a conclusion. The work contains links to 40 sources, 28 illustrations and 24 tables.

The bulk of the work is set out on 92 pages.

### **Actuality of theme**

The relevance of this work is explained by the existing problem of determining the types of objects in water in hydroacoustics.

This work is devoted to the development of a software application for calculating hydroacoustic signature with processing in Matlab.

**The purpose of the study** is the creation of a software application for determining the acoustic signature.

To achieve this goal, you need to complete the following research objectives:

- analyze the relevance of the use of acoustic signals;
- analyze similar software systems and their shortcomings;
- calculation of parameters of processing of a hydroacoustic signature;
- create, connect, and record calculation results to the database;
- develop user interface;
- develop means of graphical representation of the obtained results;

**Object of study:** Principle of processing of a hydroacoustic signal, approaches and algorithms of calculation.

**Subject of study:** Hydroacoustic signal processing, analysis of the obtained results.

**Research Methods:** Methods of mathematical and physical modeling.

**Scientific novelty:** The first comprehensive system for calculating the hydroacoustic signature, which allows to store the processed signals and compare them.

**Practical value:** The calculation of the acoustic signature will allow the users of this system to analyze the thickness of the sea or ocean, to perform the analysis to determine the neighboring object.

**Testing:** The results of research on the system are generally presented at the 5th International Scientific Conference on Sustainable Development of the 21st Century: Governance, Technology, Models (Igor Nedin Scientific Readings) 2018.

**Keywords:** SIGNAL PROCESSING, MATLAB, C#, APPLICATION INTERACTION.

## ЗМІСТ

Перелік умовних позначень, скорочень і термінів.....	9
Вступ.....	10
1 Задача визначення гідроакустичної сигнатури та подальшої її обробки.....	12
1.1 Сфери застосування .....	15
1.2 Існуючі перешкоди в гідроакустиці .....	15
1.2.1 Перешкоди, пов’язані з кораблем і його рухом.....	15
1.2.2 Перешкоди, створювані шумами морського середовища.....	17
1.2.3 Перешкоди, створювані реверберацією моря.....	21
Висновки до розділу 1 .....	22
2 Аналіз проблеми визначення гідроакустичної сигнатури.....	23
2.1 Види сигналів .....	23
2.2 Механізм обробки сигналу .....	24
2.3 Методи обробки сигналів .....	27
2.4 Перетворення Фур’є.....	28
2.5 Фільтрація шуму сигналу .....	31
2.6 Порівняння існуючих систем.....	33
Висновки до розділу 2 .....	37
3 Аналіз і обґрунтування засобів реалізації взаємодії.....	38
3.1 Набір бібліотек MATLAB Compiler Runtime.....	38
3.2 Засіб MATLAB Compiler.....	39
3.3 Пакет MATLAB Builder NE .....	40
3.4 Виклик функції MATLAB з C# Client .....	41

3.5 Огляд взаємодії додатків.....	42
Висновки до розділу 3 .....	43
4 Опис використаних програмних засобів.....	44
4.1 Середовище розробки Visual Studio 2017 .....	44
4.2 Мова програмування C# .....	46
4.3 Застосунок Windows Forms.....	48
4.4 Середовище MATLAB .....	50
4.5 Огляд СКБД Microsoft SQL Server .....	53
Висновки до розділу 4 .....	55
5 Опис програмної реалізації.....	56
5.1 Створення математичної бібліотеки .....	56
5.2 Підключення бібліотеки і її використання .....	59
5.3 Робота користувача з програмною системою.....	60
5.4 Висновки до розділу 5.....	64
6 Розробка стартап-проекту.....	65
Висновки до розділу 6 .....	80
Висновки.....	83
Список використаних джерел.....	84
Додаток А.....	88



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГАР — Гідроакустична розвідка.

ГАС — Гідроакустична станція.

НК — Надводний корабель.

ПЧ — Підводні човни.

LAN — Local Area Network (локальна мережа).

АЦП — Аналого-цифровий перетворювач.

DSP — Digital Signal Processors (цифровий сигнальний процесор).

ШПФ — Швидке перетворення Фур'є.

ДПФ — Дискретне перетворення Фур'є.

ОПФ — Обернене перетворення Фур'є.

ФХН — Формування характеристик спрямованості.

ЦАП — Цифро-аналогове перетворення.

ЕОМ — Електронна обчислювальна машина.

ООП — Об'єктно-орієнтоване програмування.

SDK — Software Development Kit (набір засобів розробки).

ASP — Active Server Pages.

CLR — Common Language Runtime.

IDE — Integrated development environment (середовище розробки).

WPF — Windows Presentation Foundation (система для побудови клієнтських додатків Windows).

UWP — Universal Windows Platform (платформа, створена Microsoft).

API — Application Programming Interfaces (прикладний програмний інтерфейс).

GUI — Graphic User Interface (графічний інтерфейс).

DDE — Dynamic Data Exchange. (механізм взаємодії додатків в операційних системах Microsoft Windows і OS / 2).

COM — Component Object Model.

## ВСТУП

У даний час можна відзначити значний інтерес до питань обробки сигналів у системах, до яких можна віднести й гідроакустичні системи. Важко уявити розробку різного роду програмних продуктів без використання математичних обчислень. Також для отримання конкретних результатів, як правило, необхідні певні маніпуляції з вхідними й вихідними даними.

Саме тому для автоматизації математичних, інженерно-технічних і наукових розрахунків необхідно використовувати різноманітні обчислювальні прилади та високотехнічні засоби — від спеціально програмованих мікрокалькуляторів до надпотужних супер-комп'ютерів. І все одно, здебільшого, такі розрахунки багато для кого залишаються складними.

У даній роботі показано практичну значущість комп'ютерного моделювання для досліджень у сфері гідроакустичних процесів, яке дає можливість спростити й частково подолати перешкоди під час роботи і розв'язати велику кількість завдань, виходячи із заданих умов середовища.

Мета виконання дипломної роботи полягає у визначенні гідроакустичної сигнатури та її подальшої обробки, збереження результатів шляхом побудови взаємодії програмних застосунків.

При розробці враховуються сильні сторони кожного з програмних застосунків для більш ефективної декомпозиції поставленої задачі.

При дослідженні було розглянуто ряд існуючих засобів обробки гідроакустичної сигнатури та інструментарій для створення взаємодії й передачі даних між додатками.

Описані вище міркування доводять актуальність поставленої задачі, бо поєднують у собі не тільки необхідність реалізації інструментарію з точки зору математичних обчислень, а й зручного графічного інтерфейсу для подальшого зберігання й використання результатів обробки, що дасть можливість “відкрити” доступ до складних математичних обчислень більшій кількості користувачів.

У першому розділі записки сформульовано постановку задачі й розглянуто перешкоди на шляху до обробки сигналу; у другому розділі проаналізовано існуючі технології, методи обробки та зроблено порівняння з існуючими системами; третій розділ містить аналіз обраних засобів, розглянуто програмну реалізацію взаємодії застосунків MATLAB і C#; у четвертому розділі описано використовувані програмні засоби; у п'ятому розділі описано методику роботи користувача; у шостому розділі описано ідею просування розробленої технології.

# 1 ЗАДАЧА ВИЗНАЧЕННЯ ГІДРОАКУСТИЧНОЇ СИГНАТУРИ ТА ПОДАЛЬШОЇ ЇЇ ОБРОБКИ

Ідея сигнатурної обробки не нова — використовуючи даний принцип, спочатку намагалися побудувати системи розпізнавання різних об'єктів. Робота системи полягає в тому, що база радар-детектора зберігає в собі інформацію характеристик випромінювання “корисних” об'єктів, наприклад, поліцейських радарів і перешкод.

Такою інформацією може бути тривалість самого імпульсу або паузи між ними, а також періодичність повторення імпульсів від конкретного пристрою. Саме ця інформація називається сигнатурою — інформація в прив'язці до конкретного устаткування, наприклад, датчика, радара, системи автомобіля чи охоронна сигналізація.

Після додавання такої інформації в пам'ять детектора він зможе оперативно виконувати аналіз характеристик випромінювання і виявляти безпосередньо дані швидкості максимально точно.

Заміряти випромінювання можна за двома принципами:

- аналізувати роботу близько розташованих радарів, що дасть можливість легко виявляти сигнали від типових пристроїв;
- досліджувати випромінювання побутових пристроїв.

Після порівняння сигналів, які прилад отримує в результаті аналізу, з наявними в пам'яті сигнатурами він видає результуючі дані у вигляді повідомлення. У цьому і полягає принцип роботи пристрою.

Поданий вище приклад описує принцип роботи сигнатурного детектора, використання якого відбувається на поверхні. Але якщо розглядати цей процес з боку гідроакустичної обробки сигнатури, то суттєвих відмінностей майже немає.

Розглянемо практичну цінність використання гідроакустичної сигнатури. Освоєння морських територій вимагає ефективних методів виявлення підводних технічних об'єктів. Вирішальна роль у виявленні відводиться пасивній

гідроакустиці, в основі якої лежить обмежене використання гідрофонів як стаціонарних, так і рухомих. Сучасна продуктивність бортової обчислювальної апаратури вже не забезпечує достатній рівень автоматизації під час прийняття рішень існуючими алгоритмами виявлення сигналів [1].

Основною ознакою присутності підводного технічного об'єкта в акваторії є типова для даного класу акустична шумова сигнатура, яка сильно змінюється як від параметрів морського середовища, так і від характеру переміщення об'єкта. Частотна і просторова неоднорідність загасання звуку в морському середовищі, зниження помітності сучасних підводних технічних об'єктів, а також повільна зміна сигналу, пов'язані з низькою швидкістю переміщення, значно ускладнюють задачу оперативного виявлення класичним способом перевищення порога при мінімальному накопиченні інформації.

Це змушує використовувати додаткові ознаки присутності об'єкта в спостережуваному сигналі, наприклад, рівень дисперсії і показник швидкості збільшення інтенсивності сигналу.

Наприклад, є пристрій, призначений для розпізнавання морських суден за їхнім шумо-випромінюванням, який досліджує спектр шумового сигналу морського судна. У досліджуваному спектрі сигналу він знаходить частоту гармоніки максимальної амплітуди і робить припущення, що це — основна частота гвинтового звукоряду [2].

Одним з основних джерел первинного гідроакустичного поля морських суден є гвинти, які створюють вібрації на двох дискретних частотах:

- вальна частота;
- лопатева частота.

Вальна частота — це частота, яка відповідає швидкості обертання валу. Лопатева частота — це частота, яка дорівнює добутку частоти обертання валу і кількості лопатей гвинта [3].

У сучасних суден вальні частоти лежать в інтервалі від 2 до 6 Гц, а лопатеві частоти від 6 до 24 Гц [4]. Внаслідок цього і в силу нелінійних ефектів, що відбуваються при випромінюванні, в низькочастотному спектрі шуму суден,

спостережуваному гідроакустичними засобами, серед гармоніки з близькими амплітудами, які становлять суцільну частину спектра, утворюється сукупність гармоніки більшої амплітуди з кратними частотами.

Дискретними складовими називають гармоніки, у яких амплітуда перевищує суцільну частину спектра. Сукупність дискретних складових з кратними частотами називається звукорядом. Звукоряд, зумовлений вібраціями гребного гвинта, називається вально-лопатевим звукорядом. В єдиному вально-лопатному звукоряді можна виділити дискретні складові вального звукоряду і дискретні складові лопатного звукоряду. Основна (перша по осі частот) частота вального звукоряду дорівнює частоті (швидкості) обертання валу, яка, в свою чергу, встановлює швидкість руху судна [2, 5]. Основна частота лопатного звукоряду дорівнює добутку частоти обертання валу і кількості лопатей гвинта.

Таким чином, вально-лопатевий звукоряд містить інформацію про кількість лопатей гребного гвинта, яку використовують в системах розпізнавання цілей, заснованих на особливостях спектрального складу сигналу [6].

У зв'язку з тим, що в сучасних гідроакустичних засобах вирішальна роль в ухваленні остаточного рішення належить людині, оцінка кількості лопатей гвинта проводиться оператором на підставі пропонованої інформації про параметри спектра сигналу, що містить вально-лопатевий звукоряд [7].

Процес визначення морського об'єкта залежно від його шумо-випромінювання зображено на рисунку 1.1.

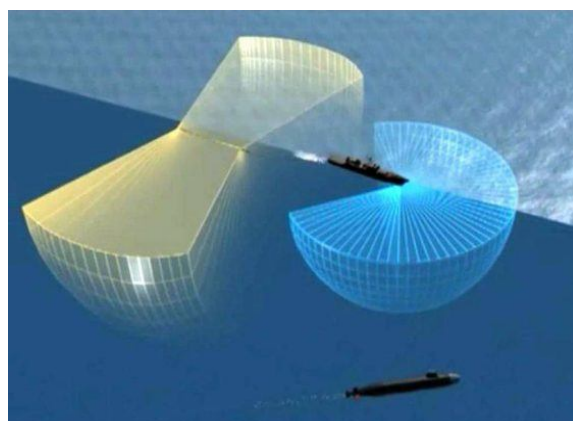


Рисунок 1.1 — Сучасні ГАС огляду підводної обстановки

## 1.1 Сфери застосування

Однією зі сфер практичного застосування гідроакустичної обробки сигналу є гідроакустична розвідка (ГАР). Під гідроакустичною розвідкою розуміється отримання інформації шляхом прийому й аналізу акустичних сигналів інфразвукового, звукового і ультразвукового діапазонів, що поширюються у водному середовищі від надводних та підводних об'єктів [8].

При веденні пасивної ГАР використовують шумопеленгатори, які приймають і аналізують шумові акустичні випромінювання у водному середовищі, які, в свою чергу, виникають при роботі двигунів, гребних валів, механізмів і машин різних агрегатів надводних кораблів (НК), підводних човнів (ПЧ) та інших плавзасобів, а також засоби розвідки, призначені для прийому й аналізу акустичних сигналів, створюваних гідролокаторами, ехолотом, системами гідроакустичного зв'язку тощо [9].

## 1.2 Існуючі перешкоди в гідроакустиці

Прийом гідроакустичних сигналів завжди здійснюється на фоні перешкод. Під гідроакустичними сигналами розуміються шуми пеленгованих цілей, ехосигнали від об'єктів пошуку, сигнали-запити та сигнали-відповіді, сигнали звуко-підводного зв'язку і розпізнавання.

Очевидно, що для отримання більш точних результатів необхідно забезпечити збільшення прийому корисних сигналів. Цьому заважають перешкоди, рівень яких залежить від різних факторів. Розглянемо ці фактори і оцінимо, як вони впливають на структуру сигналу.

### 1.2.1 Перешкоди, пов'язані з кораблем і його рухом

Перешкоди, пов'язані з кораблем-носієм і його рухом, поділяють на шумову (гвинтову), вібраційну та гідродинамічну складові. Джерелами шумової

перешкоди є валло-лопатеві кавітаційні процеси, які виникають при роботі гребних гвинтів. Шумова перешкода поширюється у воді і потрапляє на гідроакустичні антени.

Вібраційна складова утворюється працюючими механізмами корабля. Саме завдяки їй виникають вібрації корпусних конструкцій, які випромінюються в воду або передаються безпосередньо на елементи прийомних антен через корпусні конструкції, викликаючи перешкоди в них.

Гідродинамічна складова виникає внаслідок турбулізації граничного шару на поверхнях обтічників корабля, на буксирі тіла, обшивках корпусу корабля. Пульсації гідродинамічного тиску в прикордонних шарах корабля і на буксирі тіла викликають вібрації їхніх поверхонь, наслідком чого є виникнення перешкод.

Причина виникнення гвинтових перешкод полягає в тому, що при обертанні гребного гвинта надводного корабля (НК) на кінці його лопатей утворюються області зниженого тиску. Ці області наповнюються повітрям і парою води, утворюючи кавітаційні бульбашки.

Під час руху НК і при досить великій швидкості обертання гвинта ці бульбашки при переході в зони нормального тиску починають лопатися, створюючи потужний кавітаційний шум [6, 10, 11, 12] в широкому діапазоні частот. Початок кавітації на гвинті НК залежить від кількості обертів валу.

За своєю інтенсивністю корабельні перешкоди є найбільш потужними порівняно з усіма іншими видами перешкод.

У свою чергу, гвинтові перешкоди порівняно з гідродинамічними і вібраційними перешкодами є більш інтенсивними, і саме вони обмежують швидкість ходу НК.

Траєкторії поширення акустичних перешкод, які заважають, подано на рисунку 1.2, де:

- 1 — корабель-носій;
- 2 — поверхня моря;
- 3 — підкільна антена;
- 4 — корабельні механізми.



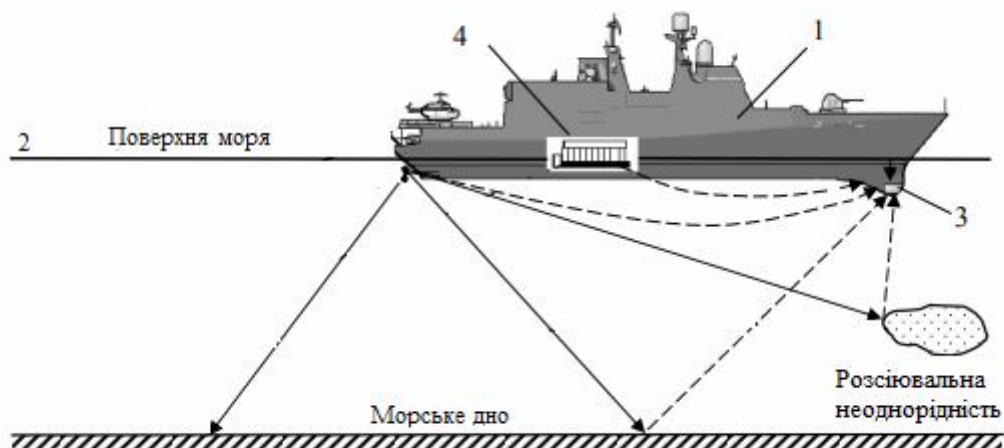


Рисунок 1.2— Траєкторії поширення акустичних перешкод

### 1.2.2 Перешкоди, створювані шумами морського середовища

До шумів морського середовища відносять такі:

- динамічні шуми, обумовлені динамікою;
- морських хвиль, турбулентних потоків у воді і атмосфері, шумом прибою, підводним шумом, шумом дощу тощо;
- біологічні шуми, створювані різними представниками морської фауни;
- сейсмічні шуми, викликані тектонічною і вулканічною діяльністю, утворенням хвиль цунамі;
- технічні шуми, які є наслідком діяльності людини, включно з перешкодами судноплавних трас;
- шуми в гаванях і прибережних районах тощо.

У будь-якій точці морського середовища шум моря є результатом накладання акустичних коливань, що приходять з різних напрямків і відрізняються за амплітудою, фазою і частотою.

Динамічні шуми відзначаються у всіх районах Світового океану за будь-яких гідрометеорологічних умов. Основними джерелами шумів в інфразвуковому діапазоні частот від  $f < 200$  до 300 Гц є турбулентні потоки у воді й атмосфері і стоячі поверхневі хвилі. Шуми штормів і циклонів, що приходять з віддалених районів, спотворюють спектр динамічних шумів. Близькість судноплавних трас і

мале загасання низьких частот зумовлює підвищення рівня шуму в діапазоні частот 20-100 Гц. У звуковому діапазоні частот шуми створюються турбулентними потоками, кавітаційними процесами, руйнуванням вітрових хвиль і дощем. Рівні шумів тісно пов'язані з гідрометеорологічними умовами даного району моря.

У цілому динамічні шуми характеризуються максимальними рівнями в зоні інфразвукових частот і їхнім спадом зі збільшенням частот — це детально зображено на рисунку 1.3 [10], де:

1 — 4 — мілке море (відповідно січень, лютий, березень, квітень);

5 — 6 — глибоке море (райони досить близькі і досить віддалені від суднохідних трас).

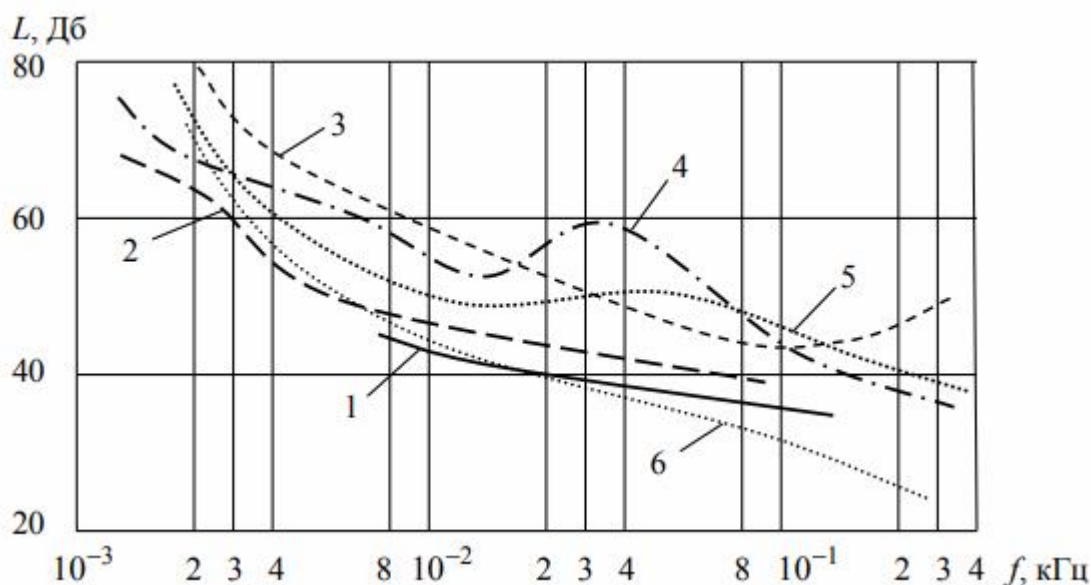


Рисунок 1.3 — Спектральні характеристики морських шумів

Особливістю динамічних шумів є також анізотропні поля шумів у вертикальній площині, характерні для області частот, більшої 200-300 Гц. Характер анізотропії істотно залежить від стану поверхні моря, розподілу швидкості звуку щодо глибини, властивостей дна тощо.

Біологічні шуми створюються живими організмами в морі, вони є численними і різноманітними. Охоплюють діапазон частот від десятків герц до десятків кілогерц (рисунок 1.4) [10].

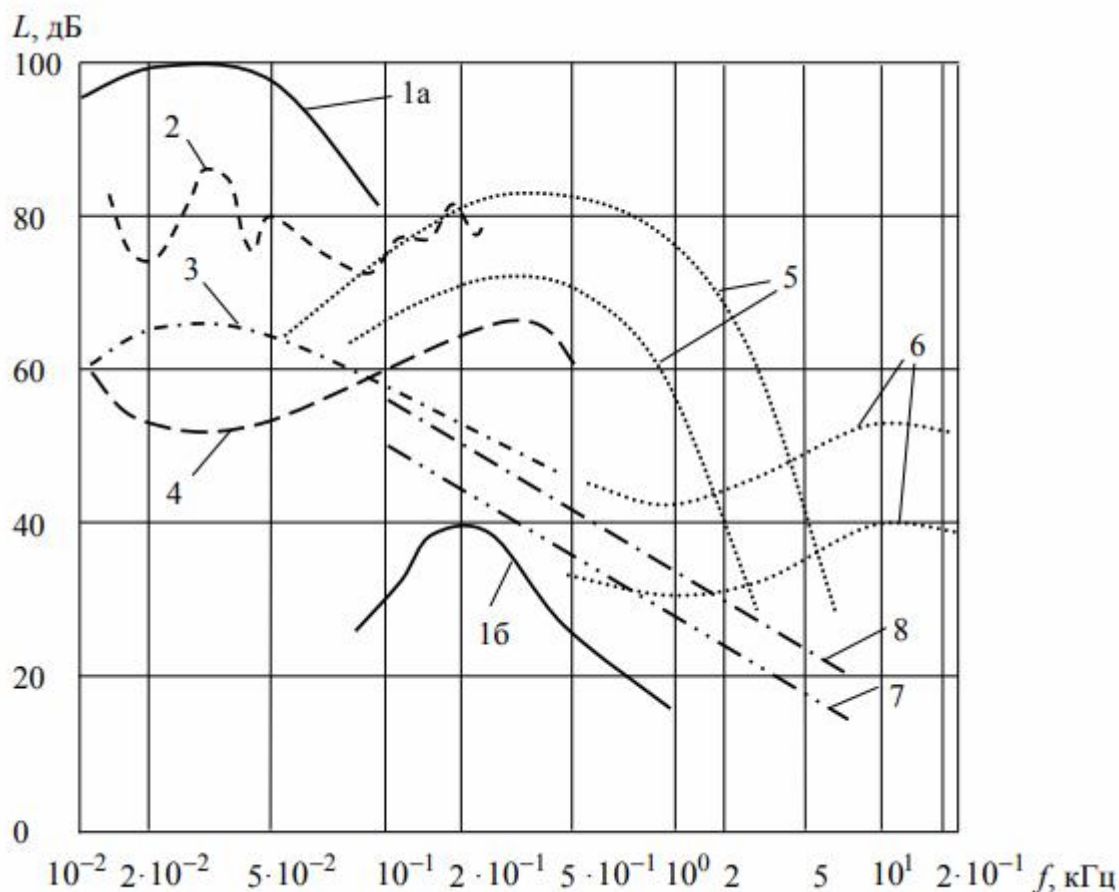


Рисунок 1.4 — Спектральна характеристика біологічних звуків і шумів

Де:

1а і 1б — спектри шуму риб і “хору” за участю риб-жаб;

2 — спектр звуку морської свині;

3 — спектр звуку морського півня;

4 — барабанщики;

5 — “хор” скупчень риб сімейства горбилевих;

6 — спектр шуму, створюваного скупченням раків-коваликів;

7, 8 — шуми моря за Кнудсенем для стану моря 2 і 4 бали.

Сейсмічні шуми. Знаходяться в області низьких частот (рисунок 1.5) [10].

Їхніми джерелами є сейсмічні процеси, які постійно відбуваються на Землі.

Джерелами інфразвукових шумів є мікросейсмічні коливання. Вони мають спектр частот від 0,1 Гц до 10 — 25 Гц. Спектр підводного шуму виверження вулкана лежить, як правило, в діапазоні частот від 1 — 3 Гц до 5 — 100 Гц.

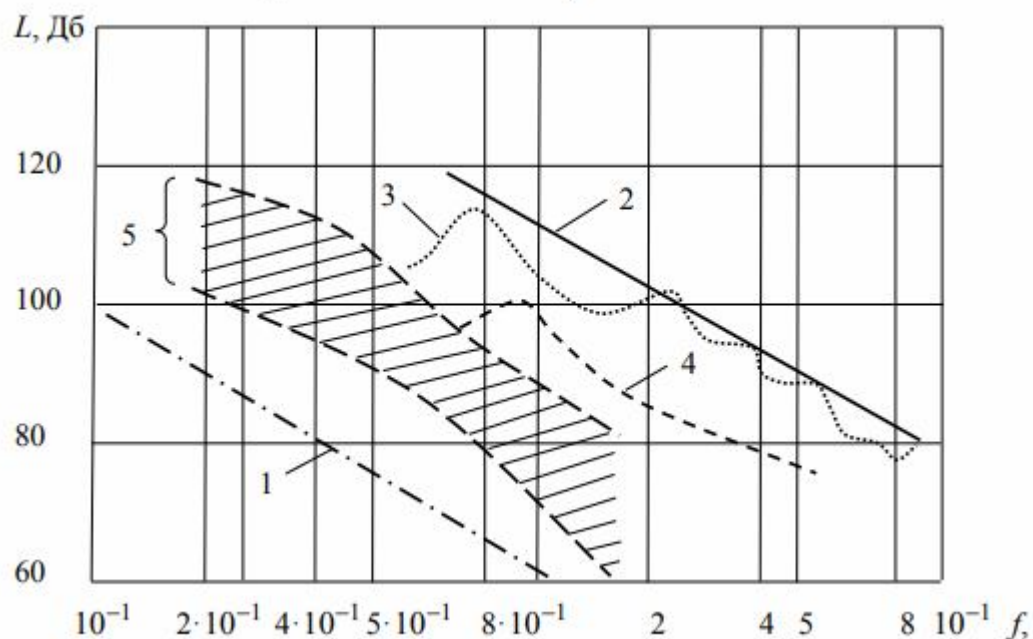


Рисунок 1.5 — Спектри шумів сейсмічного походження

Де:

- 1, 2 — мінімальний і максимальний рівні динамічних шумів;
- 3, 4 — спектри шумів далеких землетрусів у мілкому й глибокому морях;
- 5 — спектри постійного сейсмічного шуму.

Шуми судноплавства є результатом складання шумових полів великої кількості кораблів з урахуванням впливу умов поширення їх в морських умовах (рисунок 1.6) [10].

Кожен з кораблів є джерелом широкосмугового шуму. Шумові поля кораблів характеризуються наявністю суцільного спектра і дискретних складових на частотах роботи машин і механізмів та їхніх гармоніках.

З часом при рухові кораблів у спектрах їхніх шумів з'являються нові дискретні складові і суцільний кавітаційний шум. Їхні інфразвукові складові викликають підйом рівня в спектрі шумів судноплавства в діапазоні частот 20-150 Гц.

На рисунку 1.6 видно:

- 1, 2 — мінімальний і максимальний рівні динамічних шумів;
- 3 — максимальний рівень шумів у порту Нью-Йорка;
- 4, 5 — середні рівні шумів у морських портах;

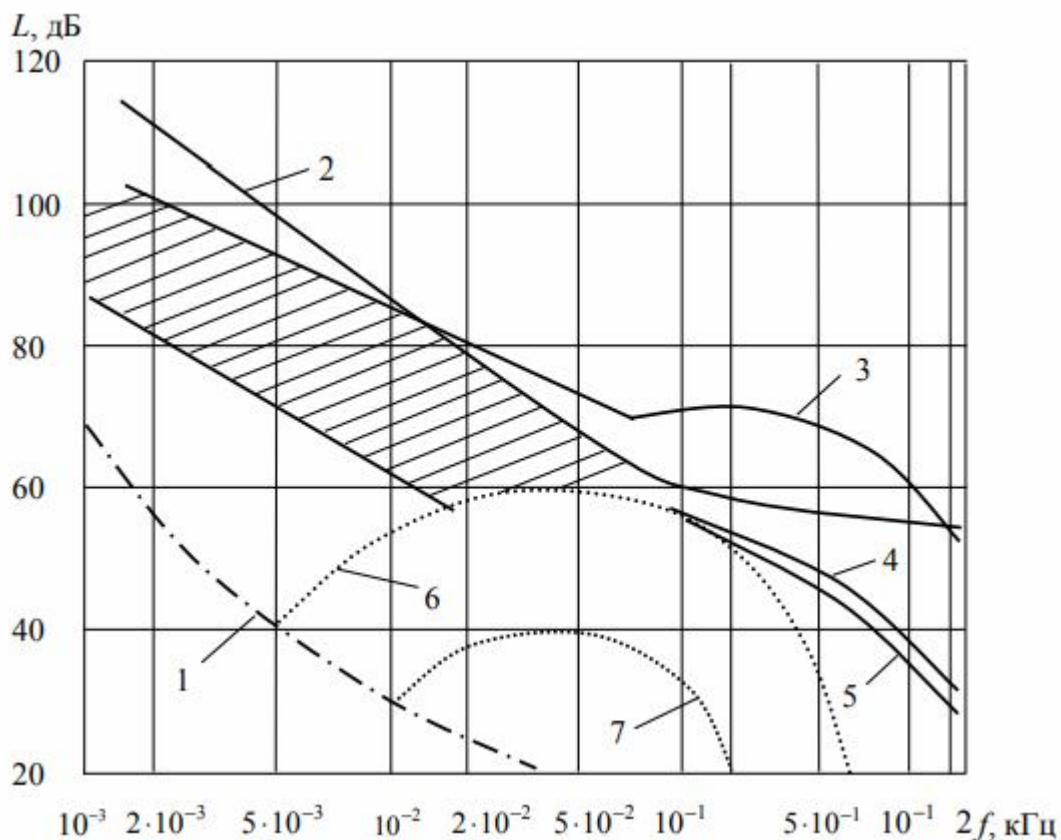


Рисунок 1.6 — Спектри шумів судноплавства

6, 7 — шуми близького і далекого судноплавства в районах трансатлантичних ліній; заштрихованная область — інфразвукові шуми в гаванях.

### 1.2.3 Перешкоди, створювані реверберацією моря

Морська реверберація — процес, що описує зміни сумарного звукового поля у часі, розсіяного морським середовищем і спостережуваного в точці розміщення антени після випромінювання зонduючого сигналу [10]. Формується на частоті випромінювання ГАС і, потрапляючи на вхід приймального тракту цієї ж ГАС, виступає в якості перешкоди виявлення сигналу від цілі. Залежно від характеру розподілу неоднорідностей, що розсіюють звук, розрізняють три види реверберації: об'ємну, шарову і граничну.

Об'ємна реверберація формується розсіювачами, що заповнюють безмежний простір.

Шарова реверберація виникає внаслідок розсіювання звуку неоднорідностями, зосередженими в шарі води. Це може бути приповерхневий шар повітряних бульбашок або розсіюючі звук шари, які знаходяться в товщі води.

## **Висновки до розділу 1**

У даному розділі було сформульовано головну задачу визначення гідроакустичної сигнатури. Описано сфери застосування даної роботи, а саме гідроакустична розвідка. Також було проаналізовано існуючі перешкоди, подано їхню класифікацію і специфіку виникнення.

## **2 АНАЛІЗ ПРОБЛЕМИ ВИЗНАЧЕННЯ ГІДРОАКУСТИЧНОЇ СИГНАТУРИ**

Для створення якісного програмного продукту треба розглянути існуючі види сигналів, механізми прийому, обробки і зберігання гідроакустичного сигналу. Виявити, які труднощі можуть виникнути для різних випадків і проаналізувати можливі варіанти подолання перешкод. Також треба розглянути й проаналізувати існуючі аналоги.

### **2.1 Види сигналів**

Сигнал визначається як напруга або струм, який може бути переданий як повідомлення або як інформація. За своєю природою всі сигнали є аналоговими, незалежно, чи це сигнал постійного або змінного струму, цифровий або імпульсний. Проте прийнято враховувати відмінності між аналоговими й цифровими сигналами. Як правило, цифрові сигнали пов'язані з реальними аналоговими сигналами, але іноді між ними немає зв'язку. Як приклад можна розглянути передачу даних у локальних обчислювальних мережах (LAN) або в інших високошвидкісних мережах.

Усі сигнали за способом подання можна розділити на чотири групи:

- аналогові — описуються неперервними в часі функціями;
- дискретні — перериваються в часі з кроком, заданим дискретизацією;
- квантовані — мають набір кінцевих рівнів (як правило, за амплітудою);
- цифрові — комбінація властивостей дискретних і квантованих сигналів.

На рисунку 2.1 подано приклади сигналів.

Цифровим сигналом є сигнал, певним чином оброблений і перетворений у цифри. Цифрові сигнали формуються з аналогових операцій дискретизації —

послідовним квантуванням (виміром) амплітудних значень сигналу через певні інтервали часу  $Dt$  або будь-яку іншу незалежну змінну  $Dx$ .

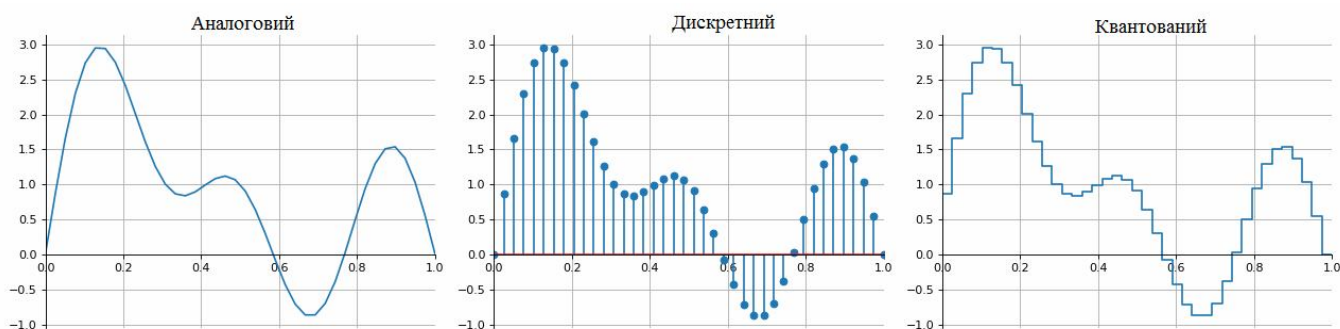


Рисунок 2.1 — Види сигналів

У результаті рівномірної дискретизації неперервний за аргументом сигнал зводиться до упорядкованої за незалежною змінною послідовності чисел.

## 2.2 Механізм обробки сигналу

Розглянемо загальну структуру обробки сигналу в тракті прослуховування, подану на рисунку 2.2.

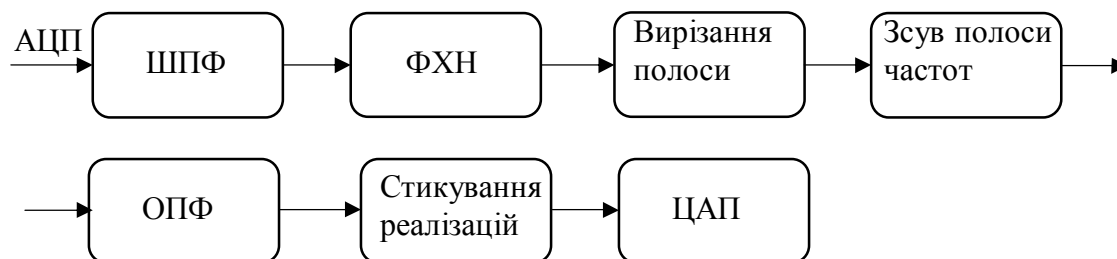


Рисунок 2.2 — Структура обробки сигналу в тракті прослуховування

У разі цифрової обробки сигналу (ЦОС) аналоговий сигнал перетвориться в двійкову форму пристроєм, який називається аналого-цифровим перетворювачем



(АЦП). Як правило, АЦП використовує двійкову систему числення з певною кількістю розрядів у рівномірній шкалі.

Збільшення кількості розрядів підвищує точність вимірювань і розширює динамічний діапазон вимірюваних сигналів. Втрачена інформація через нестачу розрядів АЦП невідновна і існують лише оцінки похибки “заокруглення” відліків. Наприклад, через потужність шуму, яка породжується помилкою в останньому розряді АЦП.

Для цього використовується поняття “сигнал / шум” — відношення потужності сигналу до потужності шуму (в децибелах). Найчастіше застосовують 8-, 10-, 12-, 16-, 20- і 24-розрядні АЦП. Кожен додатковий розряд покращує відношення сигнал / шум на 6 децибел. Однак збільшення кількості розрядів знижує швидкість дискретизації і збільшує вартість апаратури.

На виході АЦП одержується двійкове подання аналогового сигналу, яке потім обробляється арифметичним цифровим сигнальним процесором (DSP).

Після обробки інформація, яка міститься в сигналі, може бути перетворена назад в аналогову форму з використанням цифро-аналогового перетворювача (ЦАП).

Антенна з прийомним пристроєм приймає вхідний сигнал шумовипромінювання об'єкта, потім передає його до блоку АЦП, де аналоговий сигнал перетворюється в цифровий вигляд. Далі набори дискретизованих відліків фіксованої довжини послідовно в часі передаються до блоку ШПФ, який реалізує послідовне визначення спектрів прийнятого вхідного дискретизованого сигналу.

Дискретні сигнали утворюються шляхом множення аналогового сигналу  $x(t)$  на так звану функцію дискретизації  $y(t)$ , яка є періодичною послідовністю коротких імпульсів, які йдуть з певним кроком дискретизації (рисунок 2.3) [13].

В ідеальному випадку функцією дискретизації є періодична послідовність дельта-функцій.

Формування характеристик спрямованості антени і подальша обробка процесів можливі як в тимчасовій, так і в спектральній області [14]. Гідролокатор

повинен виявляти підводні об'єкти на тлі природних і штучних перешкод і супроводжувати об'єкт на відстані від 0,1 до 2-4 км. Далі відбувається зрушення смуги частот, потім зворотне швидке перетворення Фур'є і нарешті — стикування реалізацій. Після всіх вказаних етапів відбувається цифро-аналогове перетворення.

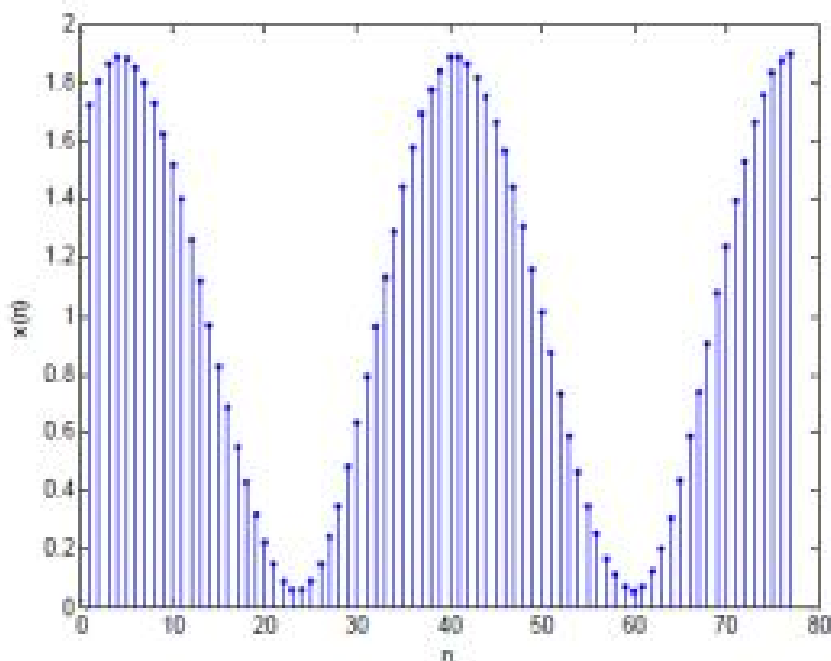


Рисунок 2.3 — Дискретизований сигнал

У ЦОС широко використовується дискретне перетворення Фур'є (ДПФ) і алгоритм його швидкого обчислення — швидке перетворення Фур'є (ШПФ). Обидва механізми дають можливість описувати всі сигнали в частотних координатах, крім миттєвих сигналів —  $< 1$  с.

Усічені за частотою Фур'є-компоненти описують дані коректніше, ніж будь-які інші степеневі ряди.

Важливим аспектом є також динамічний діапазон, який визначається максимальним і мінімальним значенням сигналу. Обробка цифрових сигналів виконується або спеціальними процесорами, або на універсальних комп'ютерах з використанням спеціальних програм.

Найпростіші для розгляду лінійні системи. Лінійними називаються системи, для яких має місце суперпозиція (відповідь на суму вхідних сигналів дорівнює сумі

відповідей на кожен сигнал окремо) і однорідність або гомогенність (зміна амплітуди вхідного сигналу викликає пропорційну зміну вихідного сигналу). Для реальних об'єктів властивості лінійності можуть виконуватися приблизно і в певному інтервалі вхідних сигналів.

Іншою ключовою концепцією у визначенні сигналу є той факт, що сигнал завжди несе деяку інформацію. Це вказує на ключову проблему обробки фізичних аналогових сигналів — проблема вилучення інформації.

## **2.3 Методи обробки сигналів**

Спектральний аналіз — один з методів обробки сигналів, який дає можливість схарактеризувати частотний склад вимірюваного сигналу.

Однією з найпоширеніших задач є виявлення в спектрі сигналу складових. Робиться це для того, щоб відфільтрувати сигнал, оскільки іноді по одній лінії передається не один сигнал, в той час як обробляти кожен з них потрібно по-різному.

Перетворення Фур'є є математичною основою, яка пов'язує тимчасовий або просторовий сигнал (або деяку модель цього сигналу) з його поданням до частотної області. Важливу роль у спектральному аналізі відіграють методи статистики, оскільки сигнали, як правило, мають випадковий характер або стають зашумлені при поширенні або вимірі. Якби основні статистичні характеристики сигналу були точно відомі або їх можна було визначити за кінцевим інтервалом цього сигналу, то спектральний аналіз був би галуззю “точної науки”. Проте в дійсності за відрізком сигналу можна отримати тільки оцінку його спектру досить суб'єктивного характеру.

Різницю між спектральними оцінками, одержуваними в результаті обробки одного й того ж відрізка сигналу різними методами, можна пояснити відмінностями припущень, прийнятих щодо даних, різними способами усереднення тощо. Якщо апріорі характеристики сигналу не відомі, не можна сказати, які з оцінок кращі.

## 2.4 Перетворення Фур'є

Одним з основних методів частотного аналізу і обробки сигналів є перетворення Фур'є. Розрізняють поняття перетворення Фур'є і ряд Фур'є.

Перетворення Фур'є передбачає неперервний розподіл частот, ряд Фур'є задається на дискретному наборі частот.

Сигнали також можуть бути задані в наборі тимчасових відліків або як неперервна функція часу. Це дає такі варіанти перетворень — перетворення Фур'є з неперервним або з дискретним часом і ряд Фур'є з неперервним часом або з дискретним часом.

Найбільш практична з точки зору цифрової обробки сигналів дискретизація і в тимчасовій, і в частотній області, але не треба забувати, що вона є апроксимацією неперервного перетворення.

Неперервне перетворення Фур'є дає можливість точно подавати будь-які явища. Сигнал, поданий рядом Фур'є, може бути тільки періодичний.

Сигнали довільної форми можуть бути подані рядом Фур'є тільки наближено, оскільки при цьому передбачається періодичне повторення розглянутого інтервалу сигналу за межами його завдання.

На стиках періодів при цьому можуть виникати розриви і злами сигналу і можуть виникати помилки обробки, викликані явищем Гіббса, для мінімізації яких застосовують певні методи (вагові вікна, продовження інтервалів задання сигналів тощо).

При дискретизації в тимчасовій або в частотній області, як правило, говорять про дискретне перетворення Фур'є (ДПФ), формула (2.1).

$$S(k) = \frac{1}{N} \sum_{n=0}^{N-1} s(n) \exp(-j \frac{2}{N} nk), \quad (2.1)$$

де  $k = 0, \dots, N-1$ ,  $N$  — кількість відліків сигналу.

Застосовується дискретне перетворення Фур'є для обчислення спектрів потужності, оцінювання передавальних функцій і імпульсних відгуків, швидкого обчислення згорток при фільтрації, розрахунку кореляції, розрахунку перетворень Гільберта тощо.

Розрахунок ДПФ за наведеною формулою вимагає обчислення  $n$  коефіцієнтів, кожен з яких залежить від  $k$  елементів вихідного відрізка, так що кількість операцій не може бути меншою від  $nk$ .

Існує ціле сімейство алгоритмів, відоме, як Швидке Перетворення Фур'є — ШПФ, що скорочує кількість операцій для обчислення коефіцієнтів до  $n \log(k)$ .

Швидке перетворення не слід трактувати, як спрощене або неточне. При точній арифметиці результати розрахунків ДПФ і розрахунків за алгоритмом ШПФ збігаються.

Перетворення Фур'є — математична основа спектрального аналізу і не тільки. Коротко розглянемо різні види перетворення Фур'є [13].

Розкладання в ряд Фур'є дає можливість розкласти неперервну функцію в суму інших неперервних функцій. У загальному випадку ряд буде мати нескінченну кількість членів.

Перетворення Фур'є неперервного в часі сигналу

$$X(f) = \int_{-\infty}^{+\infty} x(t) \exp(-2\pi f t) dt \equiv F(x(t)), \quad (2.2)$$

яке ідентифікує частоти і амплітуди тих комплексних синусоїд (експонент), на які розкладається деяке довільне коливання.

Зворотне перетворення Фур'є має вигляд:

$$x(t) = \int_{-\infty}^{+\infty} X(f) \exp(j2\pi f t) df \equiv F^{-1}(X(f)). \quad (2.3)$$

На відміну від ряду Фур'є, перетворення Фур'є розкладає функцію не за дискретними частотами (набір частот ряду Фур'є, за якими відбувається розкладання — дискретний), а за неперервними.

Спектром, у даному випадку, є коефіцієнти ряду Фур'є і результат перетворення Фур'є.

Важливо відмітити, що спектр перетворення Фур'є — у загальному випадку, функція комплексна, яка описує комплексні амплітуди відповідних гармонік.

Тобто, значення спектра — це комплексні числа, чиї модулі є амплітудами відповідних частот, а аргументи — відповідними початковими фазами. На практиці розглядають окремо амплітудний спектр і фазовий спектр.

На рисунку 2.4 видно, що коефіцієнти ряду Фур'є є ні чим іншим, як значеннями перетворення Фур'є в дискретні моменти часу.

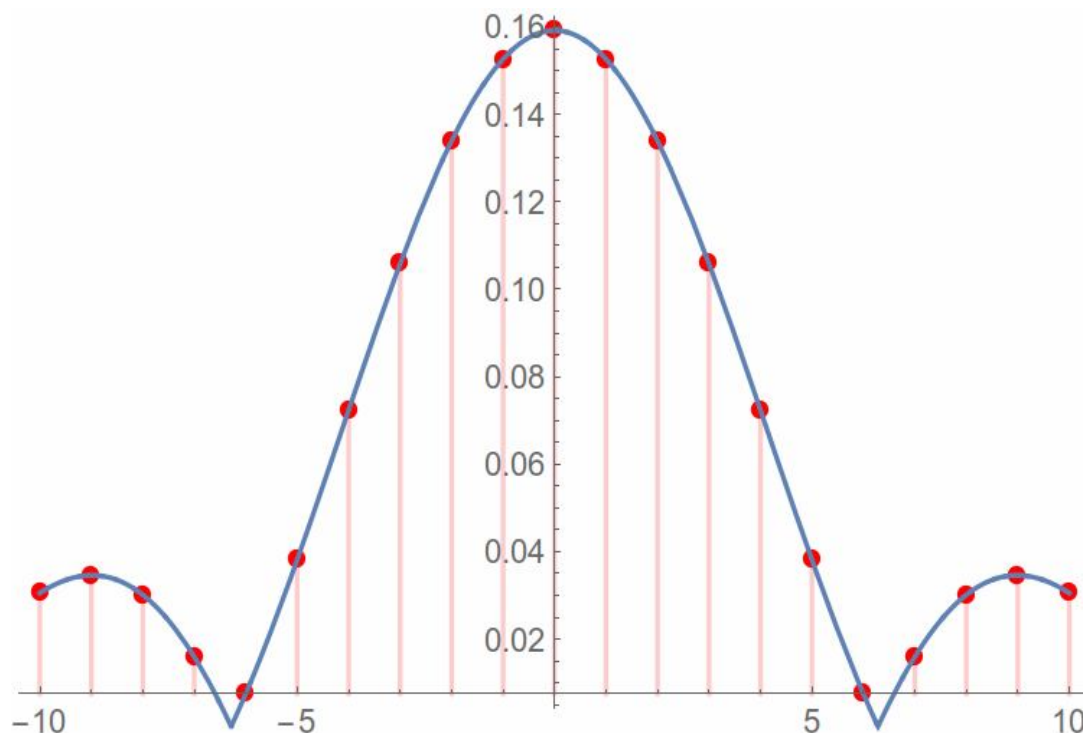


Рисунок 2.4 — Відповідність ряду Фур'є і перетворення Фур'є на прикладі амплітудного спектра

Проте, перетворення Фур'є зіставляє нескінченну функцію, неперервну у часі, іншій, неперервній за частотою, нескінченній функції — спектр.

Якщо ж немає нескінченної в часі функції, а є лише якась записана її дискретна в часі частина, то подальший розвиток перетворення Фур'є — дискретне перетворення Фур'є (ДПФ).

Дискретне перетворення Фур'є має розв'язати проблему необхідності неперервності і нескінченності сигналу в часі. По суті, ми вважаємо, що вирізали якусь частину нескінченного сигналу, а у всій іншій тимчасовій області вважаємо цей сигнал нульовим.

Математично це означає, що, маючи досліджувану нескінченну в часі функцію  $f(t)$ , множимо її на деяку віконну функцію  $w(t)$ , яка наближається до нуля всюди, крім інтервалу часу, який цікавить дослідника.

Якщо “виходом” класичного перетворення Фур'є є спектр-функція, то “виходом” дискретного перетворення Фур'є є дискретний спектр. І на вхід теж подається відлік дискретного сигналу.

## 2.5 Фільтрація шуму сигналу

Цифрова фільтрація сигналів і зображень важлива для розв'язання широкого кола науково-технічних завдань, для поліпшення якості прийому переданих повідомлень [15, 16].

Традиційно для очищення інформації від перешкод використовують перетворення Фур'є. Застосування цифрових НЧ-фільтрів у приймальному пристрої дає можливість знизити рівень адитивних шумів, що передаються сигналом, який в свою чергу передається каналом зв'язку.

Залежно від вимог, які пред'являються до інформаційних повідомлень і шумового випромінювання, також можуть використовуватися смугасто-пропускаючі або смугасто-загороджуючі фільтри [17,18].

Незважаючи на те, що математичний апарат перетворюється в ряд, він має певні обмеження.

Так, фільтри на основі перетворення Фур'є не дають можливості ефективно усувати основні особливості сигналів. У зв'язку з тим, що перетворення Фур'є використовує нескінченно осцилюючі гармонійні функції, відомості про особливості сигналу містяться у всіх коефіцієнтах перетворення, і відповідні перешкоди дуже складно відфільтрувати.

Ці проблеми частково можуть бути розв'язані в результаті застосування обраних тимчасових інтервалів.

Проте при розгляді широкосмугових процесів фіксований час не може бути локалізованим аналізом сигналу, а коригування окремих особливостей буде приводити до шуканого сигналу в деякому його околі.

Крім того, класична апаратура перетворення була розроблена для стаціонарних випадкових процесів. Це може призвести до різних проблем при інтерпретації результатів.

Для створення оптимального фільтра треба виконати постановку, формалізацію і розв'язання задачі синтезу фільтрів, які відповідають заданій сукупності показників якості та обмежень.

Головним критерієм при проектуванні таких систем, як правило, є мінімізація середньоквадратичної помилки. Залежно від того, якими рівняннями описується стан системи, оптимальні фільтри поділяються на лінійні і нелінійні.

Лінійні фільтри, які розглядаються в цьому розділі, як правило, базуються на оптимальному фільтрі Колмогорова-Вінера. Розглянемо критерії побудови оптимальних фільтрів.

На практиці обробки даних використовують два основних критерії побудови оптимальних фільтрів:

- мінімум середнього квадратичного відхилення профільтрованого сигналу від його дійсного або заданого значення;
- максимум відношення сигнал-шум і максимум енергетичного відношення сигнал-шум на виході фільтра.

При аналізі й синтезі фільтрів використовується адитивна модель вхідного сигналу, формула (2.4).

$$x(k) = s(k) + q(k), \quad (2.4)$$

де  $s(k)$  — корисна складова сигналу,  $q(k)$  — складова перешкод і шумів.



Синтез оптимальних фільтрів проводиться з максимальним використанням відомої апіорної інформації як про сигнали, які треба виділити, так і про перешкоди і шуми.

Як правило, використовується інформація про природу корисного сигналу і шуму, про їхній спектральний склад, про кореляційні і взаємні кореляційні характеристики.

Наявність певних особливостей або відмінностей у характеристиках сигналу і шуму дає можливість реалізувати фільтр загалом і також оптимальний фільтр. Якщо такі особливості відсутні, постановка завдання стає некоректною.

Визначення характеристик діючих перешкод є більш складною проблемою, але навіть при повній невизначеності можна допустити, що перешкода є нормальним стаціонарним процесом з нульовим середнім значенням.

## **2.6 Порівняння існуючих систем**

На сьогоднішній день існує досить багато систем, які здійснюють обробку гідроакустичних сигналів. Кожна з них має свої особливості, системи призначені для певної галузі, деякі налаштовані для зчитування сигналу, інші для візуалізації й моніторингу результатів обробки.

Ехосигнал від об'єкта класифікації формується на основі відображення енергії падаючого зондуєчого сигналу на об'єкт по нормалі щодо напрямку надходження зондуєчого сигналу гідролокатора. При відображенні формується регулярний фронт хвилі, характеристика якої стабільна на деякому просторовому інтервалі [19].

При цьому гідролокатор приймає відбитий ехосигнал віялом статичних характеристик спрямованості. Оскільки об'єкт знаходиться в дальньому полі і має обмежені розміри, то ехосигнал від такого відбивача є пласкою, мало перекрученою хвилею і буде прийнятий як кілька характеристик спрямованості одночасно. Ехосигнал від сукупності випадково розташованих відбивачів, які характеризують об'єкти класифікації, є випадковою сукупністю амплітуд, які будуть прийматися характеристиками спрямованості незалежно один від одного. Протяжний об'єкт

буде мати кілька відбиваючих точок, які можуть бути прийняті в сусідніх характеристиках спрямованості з різною інтенсивністю, і перетинатися вони повинні на рівні не більше ніж 0,7 від максимуму [20].

Залежно від типу об'єкта і від його протяжності кількість сусідніх просторових каналів, в яких будуть виявлені ехосигнали від об'єкта, буде різним. У цій ситуації властивості сигналу в різних просторових каналах будуть різними і будуть відрізнятися за своїми характеристиками. Якщо вимірювати протяжність об'єктів за однією характеристикою спрямованості, що має ехосигнал з максимальною амплітудою, то протяжність об'єкта буде визначатися саме цією амплітудою і не будуть враховані відбивачі, які розміщуються під іншими кутами опромінення і відображення.

Ехосигнал від об'єкта являє собою складну тривимірну картину поєднання амплітуд ехосигналу від різних точок за дистанцією, за амплітудою і за напрямком. Тому тільки сума ехосигналів від усіх відбивачів у кількох сусідніх просторових каналах буде формувати узагальнену протяжність об'єкта локації, обумовлену всіма відбивачами об'єкта. Можна отримати сумарну характеристику відбиваючих точок, що мають достатню енергію, яка апріорно невідома, у всіх просторових каналах шляхом підсумовування всіх тимчасових відліків за кількома сусідніми просторовими каналами, які формуються сусідніми характеристиками спрямованості статичного віяла. Максимальна довжина об'єкта визначить кількість сусідніх каналів, у яких буде прийматися ехосигнал, якщо об'єкт буде розміщуватися перпендикулярно до віяла характеристик спрямованості на дистанції виявлення.

Процес, зображений на рисунку 2.5, здійснюється так. Антена 1 з прийомним пристроєм приймає вхідний сигнал шумовипромінювання об'єкта, передає на блок 2 АЦП, де аналоговий сигнал перетворюється в цифровий вигляд і послідовно в часі дискретизованого відліку наборами фіксованою тривалістю передаються на блок 3 ШПФ, який виробляє послідовне визначення спектрів прийнятого вхідного дискретизованого сигналу. З виходу блоку 3 ШПФ перший спектр надходить на корелятор 5 і одночасно на накопичувач 4, з виходу якого надходить на другий вхід

корелятора, на виході якого формується коефіцієнт кореляції між першим сигналом і другим сигналом з виходу накопичувача. Отримана оцінка коефіцієнта кореляції надходить в блок 6 пам'яті. Другий спектр з виходу блоку 3 ШПФ надходить на накопичувач 4, де складається з попереднім спектром і накопичується.



Рисунок 2.5 — Схема пристрою, що реалізує розглянуте програмне забезпечення [20]

Кількість накопичень вибирається апріорно при виборі порога виявлення. Сумарний спектр подається на корелятор і визначається коефіцієнт кореляції з першим спектром, який було запам'ятовано. Вироблена оцінка коефіцієнта кореляції надходить на блок пам'яті 6 і порівнюється з першим коефіцієнтом кореляції, який було запам'ятовано, якщо він не відрізняється від вихідного коефіцієнта кореляції, то в блоці виносяться рішення, що спектри ідентичні. Таким чином, на один вхід корелятора будуть надходити спектри тимчасової реалізації

першого набору вимірювання, а на інший вхід — сумарні спектри наступних часових реалізацій.

У кореляторі відбувається вимір коефіцієнта кореляції між першим спектром і сумарними послідовними спектрами при відомій кількості накопичень, яка вибирається апіорно при виборі порога виявлення. Якщо коефіцієнт кореляції більший від порогового значення, то приймається рішення про наявність стаціонарного сигналу. Граничні значення коефіцієнта кореляції розміщуються між 0,6 і 1, що забезпечить діапазон флуктуацій коефіцієнта кореляції за рахунок випадкового впливу вхідної перешкоди і умов поширення. При плавному зменшенні коефіцієнта кореляції, викликаному зміною вхідних спектрів за рахунок впливу доплерівського зсуву частоти спектра, здійснюється корегування кількості накопичень. При цьому формується вихідний сигнал, який передається в блок 8 коригування, де відбувається зниження кількості накопичень для відновлення коефіцієнта кореляції.

Якщо збільшення коефіцієнта кореляції після коригування не відбулося, то формується сигнал про припинення використання отриманих спектральних оцінок для вироблення класифікаційних ознак і оцінки параметрів руху об'єкта до відновлення нового контакту з обраним об'єктом шумовипромінювання. Одночасно на індикатор 9 надходять спектри з виходу блоку 3 ШПФ для подання оператору і на другий вхід індикатора 9 з виходу блоку 7 надходить рішення про зміну коефіцієнта кореляції.

Антенна і приймальний пристрій 1 і аналогово-цифровий перетворювач АЦП 2 є відомими пристроями. Алгоритми визначення спектрів на основі швидкого перетворення Фур'є досить докладно описано в розділі 2.4. У сучасній гідроакустичній апаратурі сигнали, перетворені в цифровий вигляд, обробляються спеціальними цифровими процесорами на основі розроблених алгоритмів, описаних у розділі 2.3.

У процесорі реалізуються всі блоки описуваного пристрою, такі як спектральна обробка на основі ШПФ, кореляційна обробка, блоки пам'яті, процедури порівняння, накопичення, прийняття рішення і коригування.

Майже всі вказані процедури можуть бути реалізовані на сучасних комп'ютерах і ноутбуках, в яких реалізовано обчислювальні програми MATLAB, MATCAD тощо.

## **Висновки до розділу 2**

У даному розділі було розглянуто існуючі види сигналів, механізми прийому, обробки і зберігання гідроакустичного сигналу.

Розглянуто методи, які використовуються під час обробки сигналів, зокрема фільтрація та перетворення Фур'є.

Виявлено й проаналізовано можливі варіанти перешкод і шляхи їхнього подолання.

## 3 АНАЛІЗ І ОБҐРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ ВЗАЄМОДІЇ

У процесі створення програмного продукту важливо правильно обрати засоби реалізації взаємодії MATLAB і C#. Проаналізувавши основні помилки при створенні програмного забезпечення і виділивши важливі фактори для створення власного додатку, було враховано різні фактори.

Основним завданням була якість, надійність і швидкість роботи програмного продукту.

### 3.1 Набір бібліотек MATLAB Compiler Runtime

Засіб MATLAB Compiler Runtime [21] — це вільно поширюваний набір динамічних бібліотек MATLAB. Завантажити його можна з офіційного сайту і використовувати на свій розсуд.

Для використання даного варіанту необхідно завантажити MATLAB. Після встановлення створити m-файл і написати необхідну функцію. Налаштувати з'єднання через командний рядок в MATLAB, вказавши команду `deploytool`, скомпілювати. Після компіляції в папці з проектом з'явиться папка `dist`, в якій розміщується потрібна бібліотека.

Після цього необхідно додати до проекту нову бібліотеку, написати в проекті необхідний код і запустити на виконання.

З мінусів можна виділити те, що після запуску програми починають завантажуватися бібліотеки MATLAB і це відтягує запуск програми на кілька секунд. Також це є не дуже зручним способом, бо у випадку помилок у коді m-функції після виправлення помилки доведеться перекомпілювати файл і повторно додати його в проект.

Оскільки досить багато часу займає процес компілювання коду, написаного в Matlab, то виправлення всіх помилок займе певний час.

З плюсів можна виділити те, що така програма працюватиме на комп'ютері, де відсутній пакет MATLAB, але при цьому користувач отримує широкі можливості пакету MATLAB у своїй програмі.

### 3.2 Засіб MATLAB Compiler

Засіб MATLAB Compiler SDK [22] дає можливість спільно використовувати програми MATLAB в якості автономних додатків. Користувач також може створювати надбудови Microsoft Excel та інтегрувати їх в електронні таблиці Excel або пакети програм MATLAB і ділитися ними за допомогою веб-браузера. Веб-додатки на базі MATLAB розміщуються на сервері веб-додатків MATLAB, що поставляється разом з MATLAB Compiler.

Використовуючи MATLAB Compiler SDK, можна пакувати програми MATLAB в програмні компоненти для інтеграції з іншими мовами програмування.

Усі додатки, створені за допомогою MATLAB Compiler, використовують MATLAB Runtime, що дає можливість використовувати безкоштовне розгортання користувачам, яким не потрібен MATLAB.

На рисунку 3.1 зображено можливості пакету MATLAB Compiler для створення необхідних компонентів.

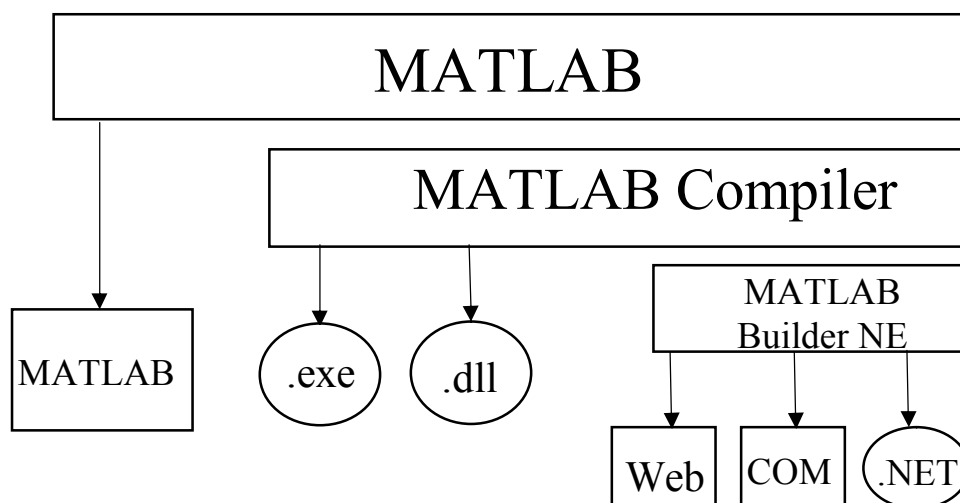


Рисунок 3.1 — Можливості пакету MATLAB Compiler

На рисунку 3.2 подано основні етапи розробки незалежного додатку для мов C/C++, але для мови C# розробка проходить аналогічно, де:

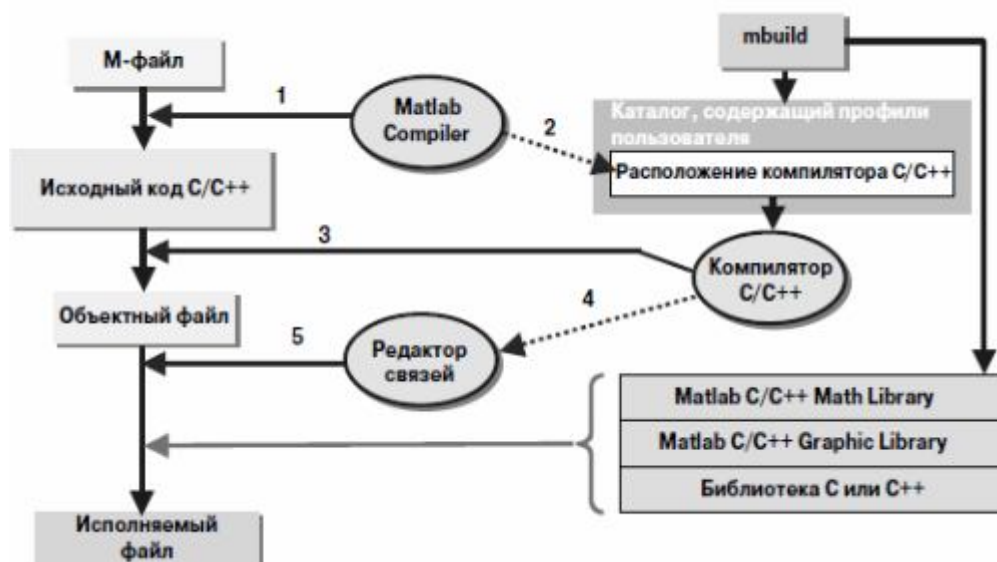


Рисунок 3.2 — Основні етапи створення незалежного додатку [23]

1 — засіб MATLAB Compiler викликає для перетворення m-код у вихідний код C#;

2 — засіб MATLAB Compiler викликає компілятор C#, використовуючи дані про його розташування;

3 — компілятор C# перетворює вихідний текст програми в об'єктний код;

4 — після завершення перетворення коду компілятор C# викликає редактор зв'язків;

5 — редактор зв'язків виконує компоновку, підключаючись до об'єктного коду необхідних бібліотек, і генерує виконуваний файл.

### 3.3 Пакет MATLAB Builder NE

Засіб MATLAB Builder NE [24] — це пакет, який розширює можливості MATLAB Compiler засобами генерації незалежних .NET-компонентів і COM-об'єктів.



Пакет MATLAB Builder NE забезпечує перетворення функцій MATLAB в незалежні компоненти, які не потребують для виконання присутності платформи MATLAB.

Поряд з можливістю генерації бібліотек, засоби створення COM і.NET-компонент забезпечують розробників потужними засобами інтеграції MATLAB-розробок з додатками, розробленими з використанням інших мов.

Компоненти, розроблені за допомогою MATLAB Builder NE, підтримують усі стандартні методи роботи з COM і .NET-об'єктами, а також спеціальні засоби перетворення різних типів даних в MATLAB-типи.

Основними особливостями MATLAB Builder NE є:

- прості інтерактивні засоби створення COM і.NET-компонент на MATLAB;
- збір і поширення незалежних компонент без додаткових ліцензійних відрахувань;
- підтримка стандартних технологій .NET для використання спільно з C, VB .NET;
- підтримка стандартних технологій .COM для використання спільно з VB, ASP;
- перетворення різних типів даних у MATLAB-типи.

### **3.4 Виклик функції MATLAB з C# Client**

Розглянемо налаштування взаємодії вікна C# з пакетом прикладних програм MATLAB [25]. Тобто, написана m-функція зберігається в каталозі MATLAB, а до створеного проекту і написаного коду C# користувач підключає через вікно Reference Manager бібліотеку. У цьому вікні треба вибрати пункт COM і підпункт Type Libraries. У списку бібліотек треба знайти бібліотеку “Matlab Application Type Library”, відмітити її і натиснути кнопку “OK”. Таким чином при запуску проекту Visual Studio також буде запускатись командний рядок MATLAB.

Саме цей спосіб є найповільнішим і для нього необхідно, щоб на комп'ютері було встановлено пакет MATLAB. Але у випадку виникнення помилок розробник

матиме можливість відразу їх виправити, не витрачаючи час на перекомпіляцію файлів і додавання бібліотек.

### 3.5 Огляд взаємодії додатків

Для реалізації взаємодії C# з MATLAB-додатком використовувалося середовище MATLAB Compiler Runtime, реалізація відбувається за алгоритмом поданим на рисунку 3.3.

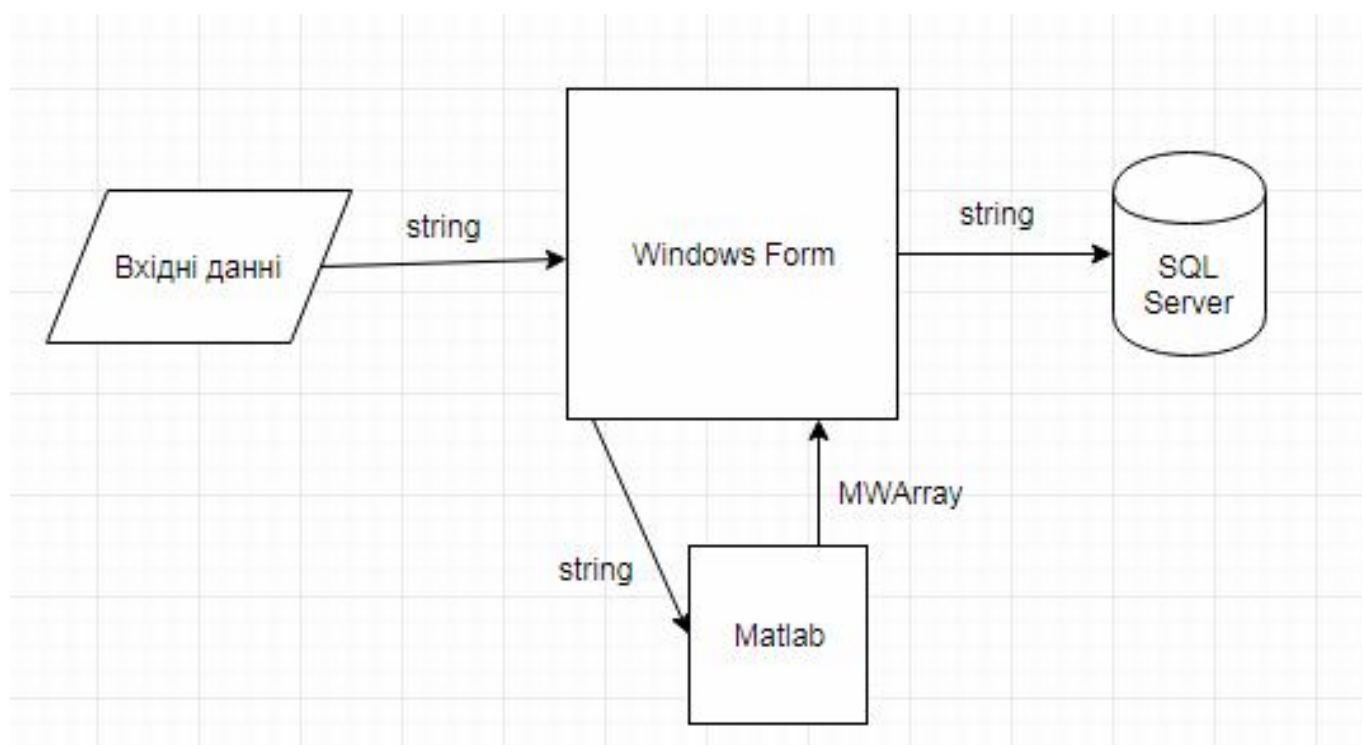


Рисунок 3.3 — Схема взаємодії C# з MATLAB

Вхідні дані до Windows Forms надходять у форматі string. Після цього дані потрапляють у MATLAB, де відбуваються розрахунки, результат яких повертається назад до форми. У самому пакеті MATLAB вхідні дані попередньо розбиваються на 4 масиви, потім фільтруються. Після цього виконується швидке перетворення Фур'є та обернене перетворення Фур'є і передбачається, що результати обробки мають виводитися графічно. Після того, як дані передаються до C#-частини і відображаються графічно, вони також записуються в базу даних.

### **Висновки до розділу 3**

У розділі проаналізовано і обґрунтовано засоби для реалізації взаємодії MATLAB і C#.

Також проаналізовано можливі підходи для досягнення поставленої мети, а саме розглянуто кілька варіантів створення необхідної взаємодії.

Розглянуто ряд переваг засобів, використаних при розробці програмного продукту для забезпечення якості, надійності і швидкості роботи програмного продукту.

## 4 ОПИС ВИКОРИСТАНИХ ПРОГРАМНИХ ЗАСОБІВ

Враховуючи тему дипломної роботи і необхідність використання сучасної мови програмування C#, зокрема для розробки графічного інтерфейсу користувача, було використано API Windows Forms, що забезпечує більш зручну реалізацію поставленої задачі.

Для розробки обчислювальної частини використовувалися базові функції математично-інженерного пакету прикладних програм MATLAB.

### 4.1 Середовище розробки Visual Studio 2017

Середовище розробки Microsoft Visual Studio 2017 [26] — це набір інструментів для створення програмного забезпечення: від планування до розробки призначеного для користувача інтерфейсу, написання коду, тестування, налагодження, аналізу якості коду і продуктивності, розгортання в середовищах клієнтів і збору даних телеметрії щодо використання. Ці інструменти призначені для максимально ефективної спільної роботи; всі вони доступні в інтегрованому середовищі розробки Visual Studio.

Середовище розробки Microsoft Visual Studio допомагає створювати додатки, що мають інноваційний інтерфейс і функціональні можливості, які зможуть задовольнити найвимогливіших замовників.

Середовище розробки Visual Studio можна використовувати для створення різних типів додатків, від простих додатків для магазину та ігор для мобільних клієнтів до великих і складних систем, які обслуговують підприємства й центри обробки даних. Користувач може створювати:

— додатки та ігри, які виконуються не тільки на платформі Windows, але і на Android і iOS;

- веб-сайти і веб-служби на основі ASP.NET, JQuery, AngularJS і інших популярних платформ;
- додатки для різних платформ і пристроїв, включаючи, але не обмежуючись: Office, Sharepoint, Hololens, Kinect і “Інтернету речей”;
- ігри й графічні додатки для різних пристроїв Windows, включаючи Xbox, з підтримкою DirectX.

На рисунку 4.1 відображено перелік проектів, які можна створити у Visual Studio 2017.

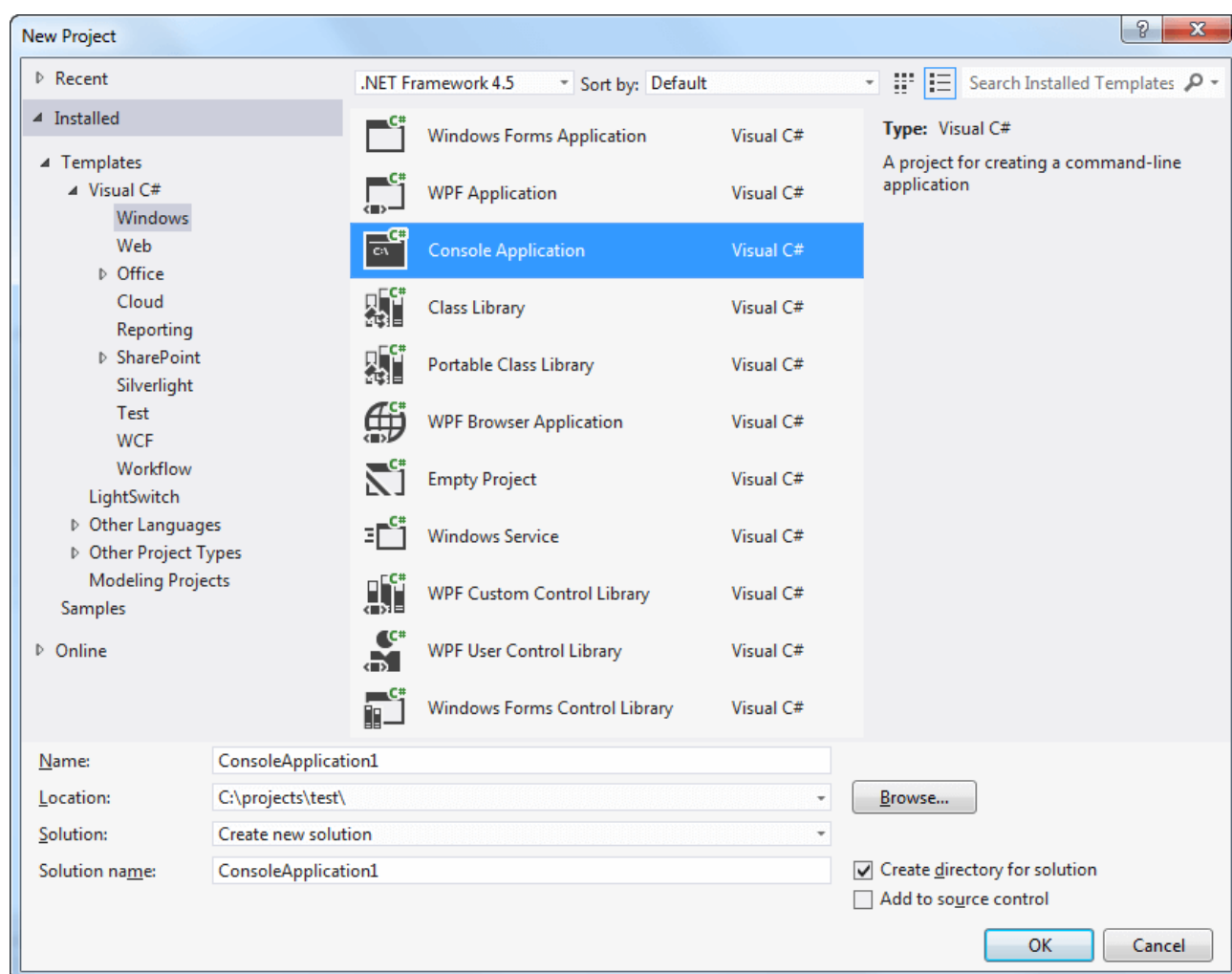


Рисунок 4.1 — Перелік проектів, які можна створити у Visual Studio 2017

За замовчуванням Visual Studio забезпечує підтримку мов програмування C#, C і C ++, JavaScript, F# і Visual Basic.

Середовище розробки Visual Studio добре працює й інтегрується зі сторонніми додатками, наприклад, Unity і Apache Cordova за допомогою розширень. Користувач також може самостійно розширити Visual Studio, створивши власні інструменти для виконання спеціалізованих завдань.

Є можливість створювати додатки для кількох пристроїв, персональних комп'ютерів і веб-додатки, використовуючи хмарні ресурси, або використовуючи вже розроблені додатки. Міжплатформні технології допомагають створювати додатки для Windows, Android і iOS за допомогою одного проекту для розробки.

Також Visual Studio пропонує потужні функції, які допомагають швидко розібратися в програмному коді. Функція кодової лупи допомагає зосередитися на роботі, показуючи посилання і зміни в коді, відображаючи автора останніх змін, а також результати проходження тестів — і все це безпосередньо в програмному коді.

## **4.2 Мова програмування C#**

Для реалізації програми було використано інтегроване середовище програмування Visual Studio Community 2017, яке безкоштовно поширюється компанією Microsoft за ліцензією для студентів, і мову програмування C#, можливості якої описано нижче. Основною мовою розробки програм на платформі .NET корпорації Microsoft є мова C#.

Мова C# — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET [31].

Мова C# розроблялась як мова програмування прикладного рівня для CLR і тому вона залежить від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 мова C# значно доповнилась. Проте ця закономірність буде порушена з виходом C# 3.0, що є розширеннями мови, що не спираються на розширення платформи .NET. Середовище CLR надає мові C#, як і всім іншим .NET-орієнтованим мовам,

багато можливостей, яких позбавлені “класичні” мови програмування. Наприклад, збірка сміття не реалізована в мові C#, а проводиться CLR для програм, написаних мовою C# так само, як це робиться для програм, написаних мовами VB.NET, J# тощо.

У мові вдало поєднуються випробувані засоби програмування з найостаннішими нововведеннями і надається можливість для ефективного і дуже практичного написання програм, призначених для обчислювального середовища сучасних підприємств.

Середовище Visual Studio дає можливість створювати програми в стилі візуального конструювання (використовуючи Windows Forms), тобто розробник оформляє свою майбутню програму і бачить результати своєї роботи ще до запуску самої програми [17]. Це дуже зручно використовувати, коли необхідно реалізувати взаємодію з програмою, яка має недоліки розробки користувацького інтерфейсу.

Технологія роботи в цьому IDE базується на ідеях об’єктно-орієнтованого й візуального програмування. В основі об’єктно-орієнтованого програмування (ООП) лежить ідея об’єднання в одній структурі даних і дій, які виконуються над цими даними [18].

Суть візуального програмування полягає в конструюванні розв’язку поставленої задачі методом вставляння компонентів (візуальних заготовок) у форму, наданні значень їхнім властивостям і в застосуванні чи створенні методів, потрібних для розв’язування задачі [19].

Процес написання додатка розділяється на дві частини: перша частина — розробник розташовує на вікно своєї програми необхідні елементи, позиціонує їх, встановлює потрібні розміри, налаштовує властивості.

Друга частина — написання програмного коду, опис властивостей елементів, доступних тільки під час роботи користувача з візуальною частиною додатка, опис реакцій на дії користувача, події вікна чи інші дії.

Середовище є одним з найбільш визнаних і популярних IDE для швидкої розробки додатків.

Технологія програмування з використанням Windows Forms значно прискорює створення додатків для настільних комп'ютерів, робочих станцій, сенсорних дисплеїв, інформаційних терміналів та Інтернету, при цьому не обмежуючи розробника в гнучкості, потужності мовних засобів або рівні контролю. Програми, створені за допомогою цього IDE, характеризуються високою швидкістю і компактністю, а також мають повнофункціональний інтерфейс користувача і здатність підлаштовуватися практично до будь-якого додатку або іншого джерела даних, не вимагаючи додаткового доопрацювання.

### 4.3 Застосунок Windows Forms

Застосунок Windows Forms — інтерфейс програмування додатків (API), що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework.

Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за рахунок створення обгортки для існуючого Win32 API в керованому коді. Windows API (application programming interfaces) — загальне найменування для цілого набору базових функцій інтерфейсів програмування застосунків операційних систем сімейства Windows корпорації Майкрософт — є найпрямішим способом взаємодії застосунків з Windows.

Для створення програм, які використовують Windows API, Майкрософт випускає SDK, який називається Platform SDK і містить документацію, набір бібліотек, утиліт і інших інструментальних засобів.

Причому керований код — класи, які реалізують API для Windows Forms, не залежать від мови розробки. Тобто програміст однаково може використовувати Windows Forms як при написанні програмного забезпечення мовами C#, C++, так і мовами VB.Net, J# тощо.

З одного боку, Windows Forms розглядається як заміна більш старої і складної бібліотеки MFC, яка спочатку була написана мовою C++. З іншого боку, Windows Forms не пропонує парадигму, порівнянну з Model-View-Controller. Для



виправлення цієї ситуації і реалізації даної функціональності в Windows Forms існують сторонні бібліотеки.

Однією з найбільш використовуваних подібних бібліотек є User Interface Process Application Block, випущена спеціальною групою Microsoft, що займається прикладами і рекомендаціями, для безкоштовного скачування. Ця бібліотека також містить вихідний код і навчальні приклади для прискорення навчання [26].

Усередині .NET Framework, Windows Forms реалізується в рамках простору імен System.Windows.Forms.

Додаток Windows Forms є подієво-орієнтованим, що підтримується Microsoft .NET Framework.

На відміну від пакетних програм, велика частина часу витрачається на очікування від користувача будь-яких дій, як, наприклад, введення тексту в текстове поле або клацання мишкою по кнопці [27].

Приклад додатку Windows Forms подано на рисунку 4.2.

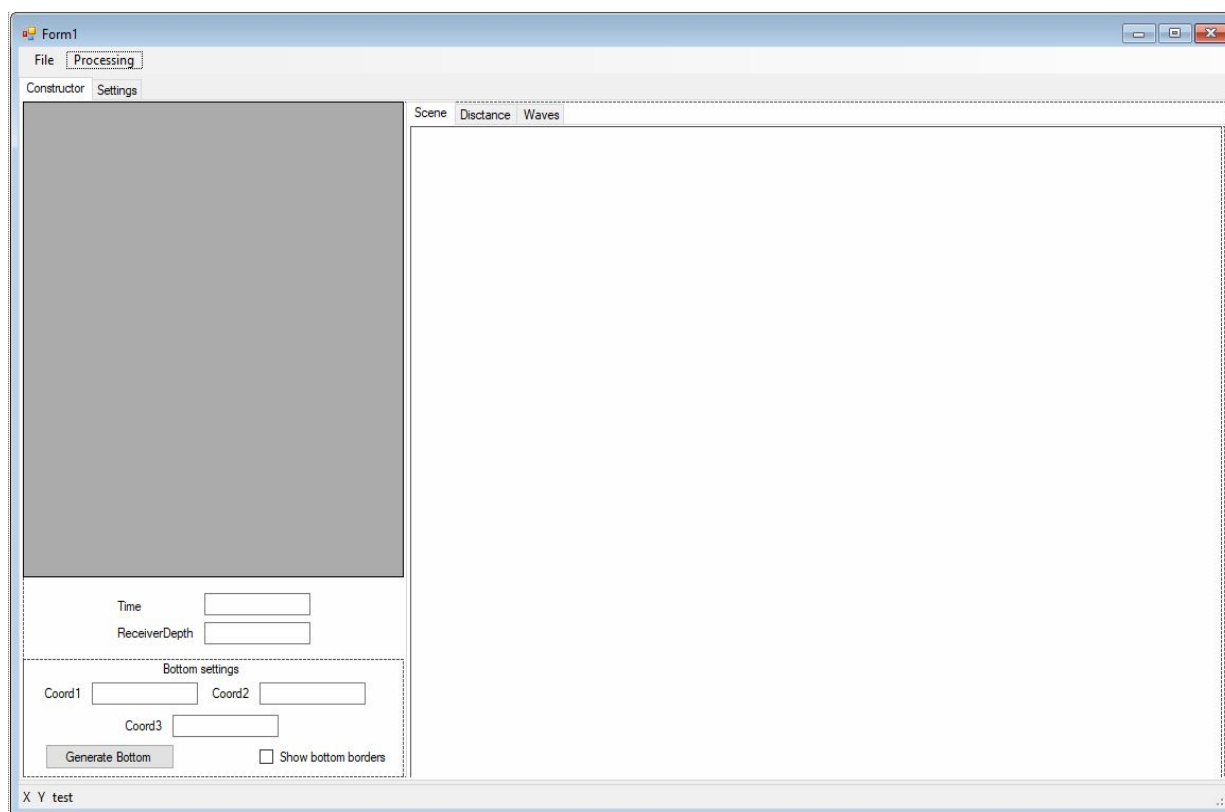


Рисунок 4.2 — Приклад Windows Forms

Для подальшого розвитку корпорація Майкрософт домоглася успіху у використанні WinForms з XAML за допомогою таких фреймворків, як WPF і UWP. Проте перетягування компонентів графічного інтерфейсу способом, подібним до WinForms, забезпечується за допомогою XAML, замінюючи кореневий елемент XAML сторінки або вікна за допомогою інтерфейсу користувача Canvas. При виконанні цієї зміни користувач може побудувати вікно аналогічним чином, як і у WinForms, шляхом безпосереднього перетягування компонентів за допомогою графічного інтерфейсу Visual Studio.

Хоча XAML забезпечувала перетягування і зворотну сумісність місць розташування за допомогою контрольної панелі Canvas, слід зазначити, що елементи керування XAML подібні до WinForm Controls і не сумісні з один одним. Вони виконують подібні функції та мають подібний зовнішній вигляд, але властивості й методи досить різні, щоб вимагати перенаправлення з одного API на інше.

## **4.4 Середовище MATLAB**

Найбільш відомим середовищем математичного програмування є продукт MATLAB (актуальна версія 2017b) компанії MathWorks. Дане рішення використовується для модельно-орієнтованого проектування (імітаційного моделювання), розробки систем управління, систем зв'язку і цифрової обробки сигналів, генерації коду для мікроконтролерів, аналізу і збору даних, а також для інших обчислювальних задач.

Виходячи з широкої сфери застосування, і з огляду на те, що MATLAB є середовищем вільного розповсюдження, виконання компонентів MATLAB MCR, завдання інтеграції функціональних блоків, реалізованих в середовищі MATLAB, в кінцевий додаток або інформаційну систему стає актуальним. Рішення такого завдання може будуватися на різних принципах взаємодії залежно від платформи розробки кінцевого додатка.

Середовище MATLAB працює на більшості сучасних операційних систем, включаючи Linux, Mac OS і Microsoft Windows. MATLAB, як мова програмування, була розроблена в кінці 1970-х років з метою полегшення процесів програмування для студентів (мова розділу Simulink отримала назву візуального проектування). Нова мова була з великим інтересом зустрінута вченими, які працюють у сфері прикладної математики. Вдосконалений варіант пакету MATLAB мовою C з'явився в 1984 році.

Середовище MATLAB призначалося для проектування систем управління, але використовується також в багатьох інших наукових і інженерних галузях. Воно також широко використовувалося і в освіті, зокрема, для викладання лінійної алгебри і чисельних методів.

Мова MATLAB є високорівневою інтерпретованою мовою програмування, яка включає засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об'єктно-орієнтовані можливості й інтерфейси до програм, написаних іншими мовами програмування.

Основною особливістю пакету прикладних програм MATLAB є його широкі можливості роботи з матрицями. Пакет MATLAB надає користувачеві велику кількість функцій для аналізу даних, що покривають практично всі розділи математики.

Пакет MATLAB надає зручні засоби для розробки алгоритмів, включаючи високорівневі з використанням концепцій об'єктно-орієнтованого програмування [28]. У ньому є всі необхідні засоби інтегрованого середовища розробки, включаючи відладчик і профайлер. Функції для роботи з цілими типами даних полегшують створення алгоритмів для мікроконтролерів та інших застосувань, де це необхідно.

У складі пакету MATLAB є велика кількість функцій для побудови графіків, зокрема тривимірних, візуального аналізу даних і створення анімованих роликів. Вбудоване середовище розробки дає можливість створювати графічні інтерфейси користувача з різними елементами управління, такими як кнопки, поля введення та іншими.

За допомогою компоненту MATLAB Compiler ці графічні інтерфейси можуть бути перетворені в самостійні застосування, для запуску яких на інших комп'ютерах має бути встановлена бібліотека MATLAB Component Runtime.

Пакет MATLAB містить функції, які надають йому доступ до інших додатків середовища Windows так само, як і цим застосуванням діставати доступ до даних MATLAB, за допомогою технології динамічного обміну даними (DDE). Інтерфейс для послідовного порту пакету MATLAB забезпечує прямий доступ до периферійних пристроїв, таких як модеми, принтери і наукове устаткування, яке підключається до комп'ютера через послідовний порт (COM-порт). Інтерфейс працює шляхом створення об'єкта спеціального класу для послідовного порту. Наявні методи цього класу дають можливість читати й записувати дані в послідовний порт, використовувати події і обробники подій, а також записувати інформацію на диск комп'ютера в режимі реального часу.

Це необхідно при проведенні експериментів, симуляції систем реального часу і для інших застосувань. Для пакету MATLAB є можливість створювати спеціальні набори інструментів, які розширюють його функціональність.

Наборами інструментів є колекції функцій, написаних мовою MATLAB для розв'язування певного класу завдань.

Однією з найпопулярніших платформ для розробки додатків є рішення компанії Microsoft — .NET Framework. Дане уніфіковане середовище розробки використовується програмістами всього світу і існує безліч рішень побудованих на даному фреймворку.

Використання даної платформи надає широкі можливості для взаємодії з операційною системою, а також набором периферійних пристроїв на низькому рівні. Можна з упевненістю говорити про те, що .NET забезпечує більшу інтеграцію додатків з ОС порівняно з внутрішніми засобами MATLAB (наприклад, MATLAB GUI Tools), що ще раз підводить до висновку про доцільність використання компонентів MATLAB при побудові додатків на базі MS .NET Framework [29].

Конструктор графічного інтерфейсу GUIDE [30] є одним з набору засобів, які надають можливість для візуально-орієнтованого програмування і проектування GUI (Graphic User Interface). Такий підхід характерний для ряду сучасних візуально орієнтованих мов програмування, таких як Visual C#, і суттєво полегшує проектування GUI [31]. Це зумовлено тим, що в даному випадку генерація програмного коду відбувається автоматично, що мінімізує зусилля користувача в плані програмування, а іноді і взагалі не потребує таких зусиль.

## 4.5 Огляд СКБД Microsoft SQL Server

Комерційна система керування базами даних Microsoft SQL Server розповсюджується корпорацією Microsoft.

Мова, яка використовується для запитів — TransacT-SQL, створена спільно Microsoft та Sybase [32].

Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних [33].

Базовий код MS SQL Server (рисунок 4.3) (до версії 7.0) ґрунтувався на коді Sybase SQL Server.

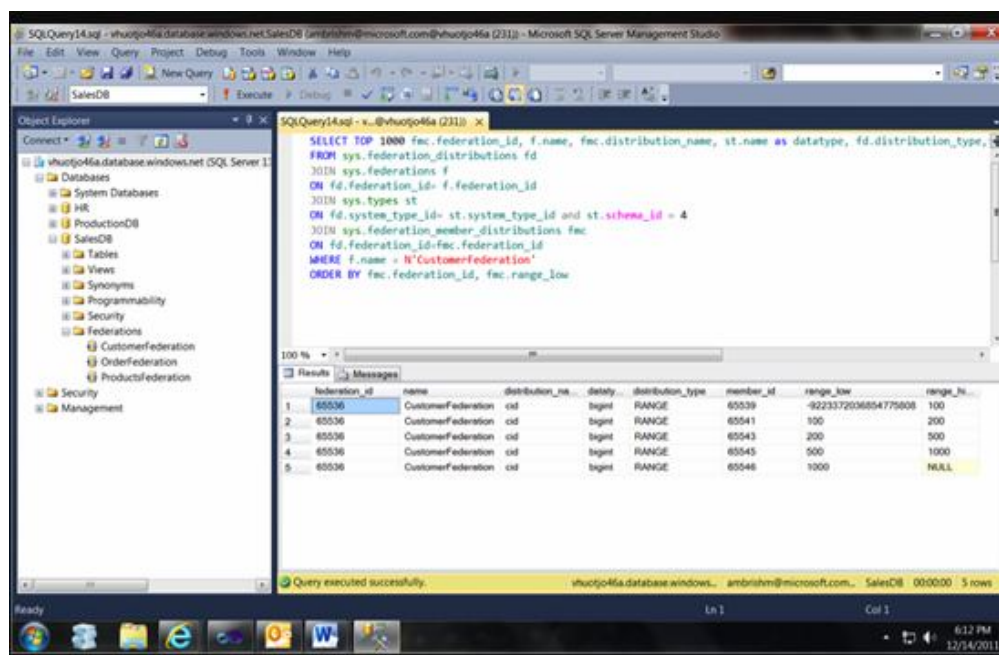


Рисунок 4.3 — MS SQL Management Studio

Середовище Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву TRANSACT-SQL яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями [34].

Мова TRANSACT-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Сервіс Microsoft SQL Server також підтримує Open Database Connectivity — інтерфейс взаємодії застосунків з СУБД [35]. Версія SQL Server 2005 надає можливість підключення користувачів через веб-сервер-сервіси, які використовують протокол SOAP [36]. Це дає можливість клієнтським програмам, не призначеним для Windows, кросплатформно з'єднуватися з SQL Server. Корпорація

Сервіс SQL Server підтримує дзеркалювання і кластеризацію баз даних. Кластер сервера SQL — це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між кількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу.

Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер [37]. Синхронізація з іншими серверами. Базы даних кількох серверів синхронізуються між собою. Зміни усіх баз даних відбуваються незалежно на кожному сервері, а під час синхронізації відбувається звірка даних. Дублювання такого типу передбачає можливість розв'язання протиріч між базами даних [38].

Сервіс SQL Server 2005 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані будь-якою мовою платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework [39]. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows [40]. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

## **Висновки до розділу 4**

У даному розділі розглянуто і описано засоби, які були використані під час розробки програмного продукту.

Розроблені додатки призначені для роботи в ОС Microsoft Windows, тому обрано відповідне середовище для роботи з мовою C#.

## 5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Розглянемо технологію інтеграції мови програмування Visual C # з MATLAB R2016a на прикладі розв’язання задачі визначення гідроакустичної сигнатури. Робота виконана в рамках розробки програмного забезпечення, призначеного для виконання завдання дипломної роботи і складається з двох етапів. На першому етапі здійснюється створення математичної бібліотеки, яка виконує обчислення задачі, а на другому її підключення до робочої програми.

### 5.1 Створення математичної бібліотеки

Важливою особливістю MATLAB порівняно з іншими математичними пакетами є наявність засобів, які дають можливість використовувати функціонал, створений в MATLAB, при розробці автономних додатків. Така можливість реалізується за допомогою продукту MATLAB Compiler. Даний продукт дає можливість збирати модулі у вигляді:

- незалежний додаток;
- C / C ++ бібліотеки (DLL в Windows, колективні бібліотеки в Linux);
- програмні компоненти такі, як класи Java, збірки .NET або надбудови Microsoft Excel для використання в інших додатках.

Додатки й бібліотеки, створені за допомогою MATLAB Compiler, використовують виконуване ядро, яке вільно розповсюджується, — так зване MATLAB Compiler Runtime (MCR). Це дає можливість запускати додатки, які використовують MATLAB-функції, без необхідності установки дорогої копії MATLAB.

Одним з інструментів для створення математичних бібліотек або зовнішніх компонентів є Deployment Tool, що представляє собою графічний інтерфейс. Виклик даного інструменту здійснюється командою `deploytool`. На рисунку 5.1 представлено



стартовое вікно Deployment Tool, на якому пропонується вибрати тип і створити новий проект.

Назва проекту визначає простір імен, що використовується в подальшому в Visual C #. Проект складається з класів, які в свою чергу включають m-файли. Кожен m-файл включає одну функцію, яка викликається як метод відповідного класу в Visual C #.

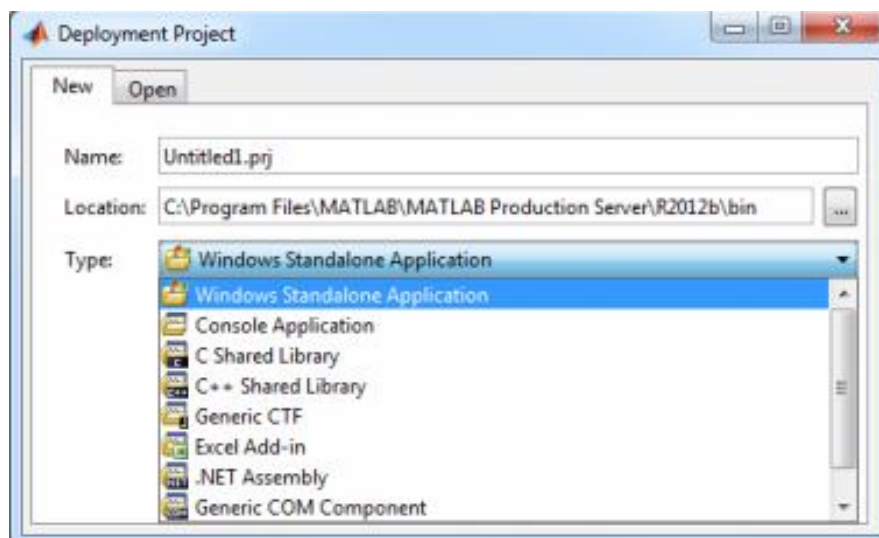


Рисунок 5.1 — Стартове вікно інструменту Deployment Tool в MATLAB

Розглянемо задачу обробки сигналу з файлу, власноруч написаної мовою MATLAB.

Файл з чотирьохканальним сигналом розміщується в певному каталозі, маючи розширення \*.dat. Файл представляє собою набір даних у шістнадцятковому вигляді, розміщених послідовно. Формат файлу з сигналом подано в таблиці 5.1.

Таблиця 5.1. Формат \*.dat-файла з сигналом

0	0		
$P_1$	$V_{x1}$	$V_{y1}$	$V_{z1}$
...	...	...	...
$P_n$	$V_{xn}$	$V_{yn}$	$V_{zn}$

де  $P$  — амплітуда тиску сигналу;  $V_x$ ,  $V_y$ ,  $V_z$  — проекції вектора коливань на осі  $x$ ,  $y$ ,  $z$ .

Для роботи з файлами у MATLAB є такі функції як:

- fread / fwrite;
- fscanf / fprintf;
- dlmread / dlmwrite;
- imread і imwrite.

Функція save дає можливість зберігати довільні змінні програми в файл, а load — завантажити із зазначеного m-файлу раніше збережені змінні.

Функції fscanf / fprintf — для роботи з файлами, дають можливість записувати і зчитувати інформацію побайтно.

Функція dlmread /— читає / записує числові дані з ASCII-файла.

Функції imread і imwrite — для роботи з файлами зображень, поданих в форматах .bmp, .png, .gif, .jpeg, .tif тощо.

Оскільки сигнал подано у шістнадцятковому вигляді, було використано функцію fscanf з властивістю “%x” для знакових шістнадцяткових цілих чисел.

Для реалізації введення та обробки вхідних даних, під час розробки програмного забезпечення, було розроблено відповідний програмний модуль засобами MATLAB, його базова реалізація полягає в тому, що він відповідає за отримання даних файлу, що лежить за переданим шляхом, враховуючи багатоканальність та дискретизацію даних у файлі з гідроакустичним сигналом і також відповідає за розбиття вхідного потоку даних на канали. Реалізація враховує той факт, що кожна точка сигналу описується двома байтами, а також те, що дані з кожного каналу чередуються в отриманому потоці.

Після зчитування сигналу його було поділено на 4 масиви:  $P$ ,  $V_x$ ,  $V_y$  та  $V_z$ . Це значення коливальних швидкостей та сумарного акустичного тиску в певний момент часу моделювання, вони і будуть характеризувати наш сигнал.

Усі операції щодо цифрової обробки сигналу було виконано за допомогою математичних функцій пакету MATLAB і подано у вигляді графіків.

Таким чином, в MATLAB був створений проект fscan з класом Signal і функцією Res. Для використання написаних функцій проект компілюється в бібліотеку, що підключається динамічно — DLL.

## 5.2 Підключення бібліотеки і її використання

Для використання зовнішньої бібліотеки середовище розробки Visual Studio 2017 і мова програмування Visual C # потребує додавання збірки.

Для підключення написаних в MATLAB функцій необхідно додати посилання на створену бібліотеку, а також на бібліотеку MWArray, що знаходиться в папці встановленої програми за адресою “C: \ Program Files \ MATLAB \ MATLAB Production Server \ R2016a \ toolbox \ dotnetbuilder \ bin \ win32 \ v2.0 \ MWArray.dll”.

На рисунку 5.2 подано вікно додавання нових посилань на збірку.

На рисунку 5.3 зображено підключення бібліотек безпосередньо у самому коді, де одна з бібліотек — це збірка \*.dll MATLAB функції, а друга бібліотека .NET — MWArray.

Бібліотека MWArray призначена для обміну даними між середовищем програмування MATLAB і платформою .NET. Вона забезпечує перетворення типів даних, відстежування помилок виконання коду, експорт графіки в додатки ASP .NET та ін.

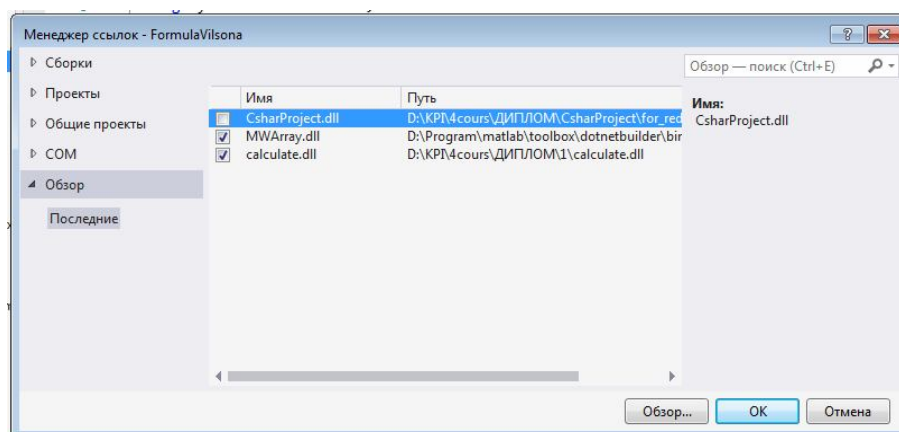


Рисунок 5.2 — Вікно додавання посилань на збірку в Visual Studio

Відзначимо обов'язковість дотримання розрядності MATLAB і створюваного програмного забезпечення.

Також помічена нестабільність роботи при роботі з 64-розрядною архітектурою, тому при розробці використовувалася 32-розрядна версія MATLAB.

Для використання бібліотеки необхідно додати опис простору імен і викликати зовнішню функцію MATLAB Class1 obj = new Class1();

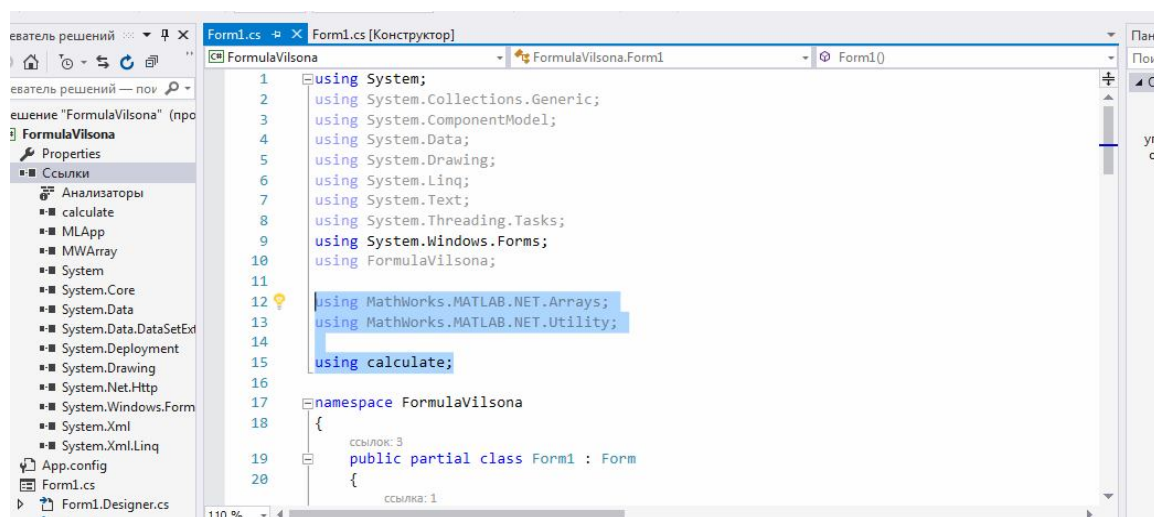


Рисунок 5.3 — Підключення бібліотек

З використанням мови C# написано програмний код створення Windows form і на неї додано необхідні елементи управління. Також розроблено й написано функцію відкриття файлу сигналу.

### 5.3 Робота користувача з програмною системою

Після запуску програми з'являється початкова форма, зображена на рисунку 5.4. Форму можна робити ширшою або вужчою.

Компоненти, нанесені на форму, будуть підлаштовуватися під розмір форми завдяки адаптивному дизайну.

Для завантаження файлу потрібно натиснути кнопку “Load file”, яка розміщується на формі.

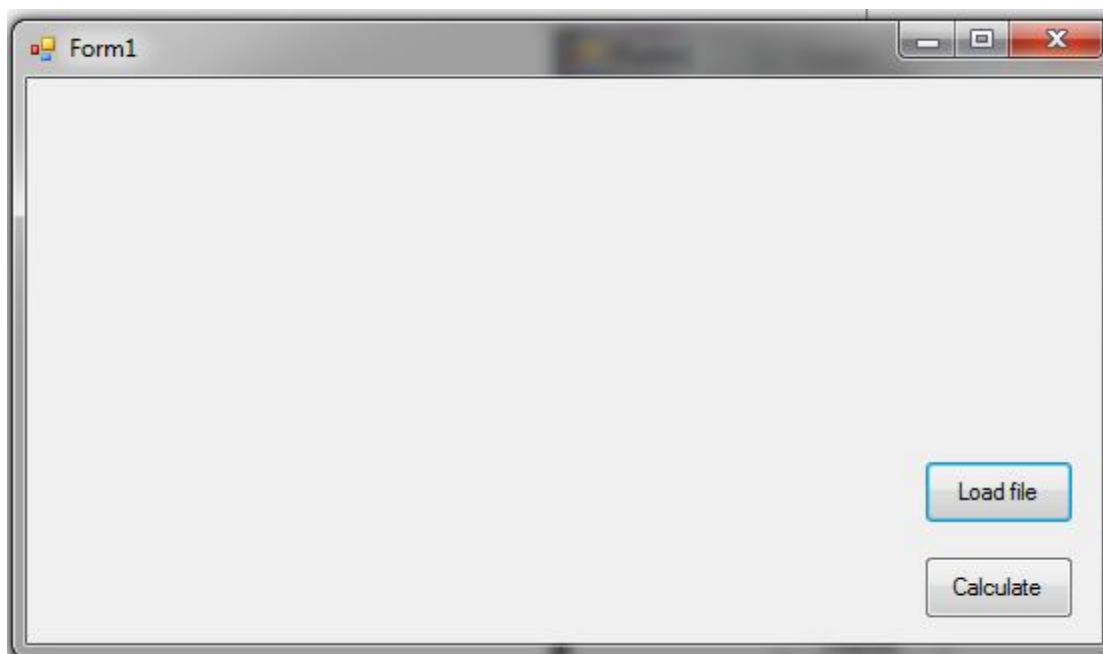


Рисунок 5.4 — Початкова форма робочої програми

Після того, як користувач натисне на кнопку, відкриється вікно каталогу з файлами, де потрібно вибрати файл для завантаження (рисунок 5.5).

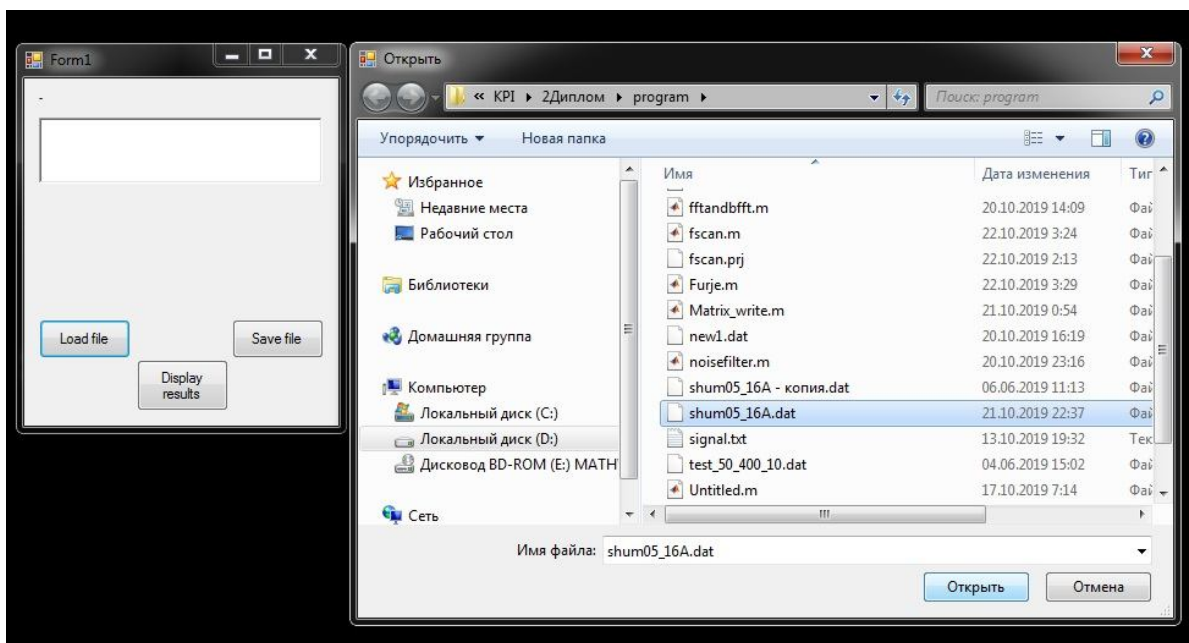


Рисунок 5.5 — Завантаження файла

За замовчуванням у програмі прописано шлях до папки, яку потрібно відкривати, але її можна змінити на будь-яку папку в комп'ютері користувача.

Після того, як користувач вибере потрібний файл, вікно каталога закриється і на формі відобразиться ім'я вибраного файла (рисунок 5.6).

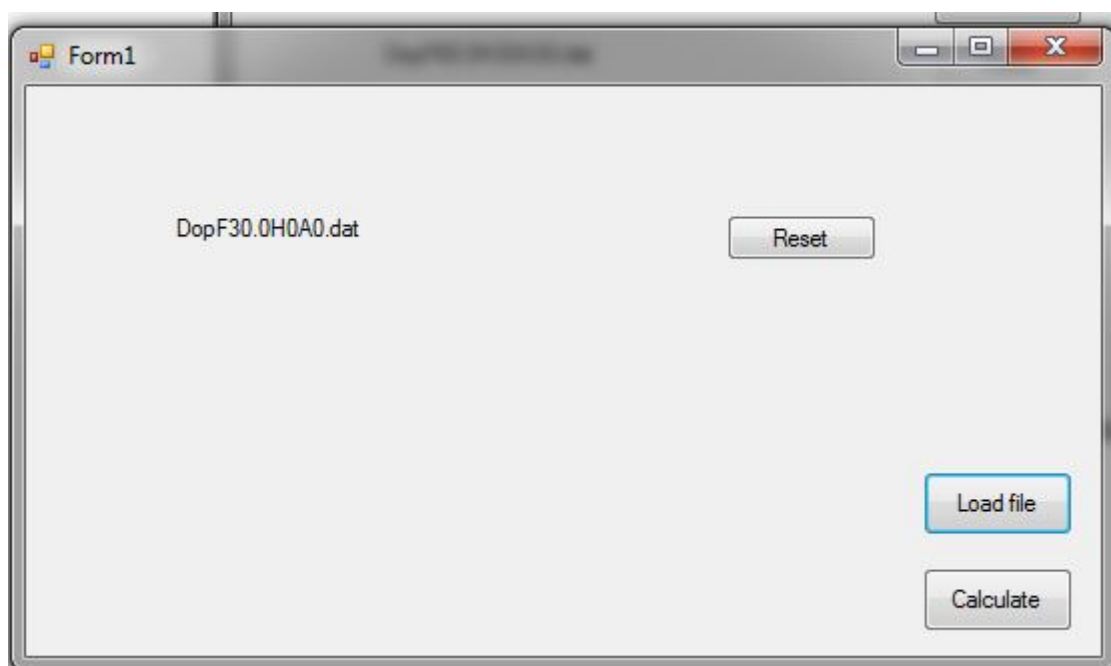


Рисунок 5.6 — Форма з завантаженням файлом

Користувач також може додати кілька файлів. Вони будуть оброблятися один за одним (рисунок 5.7).

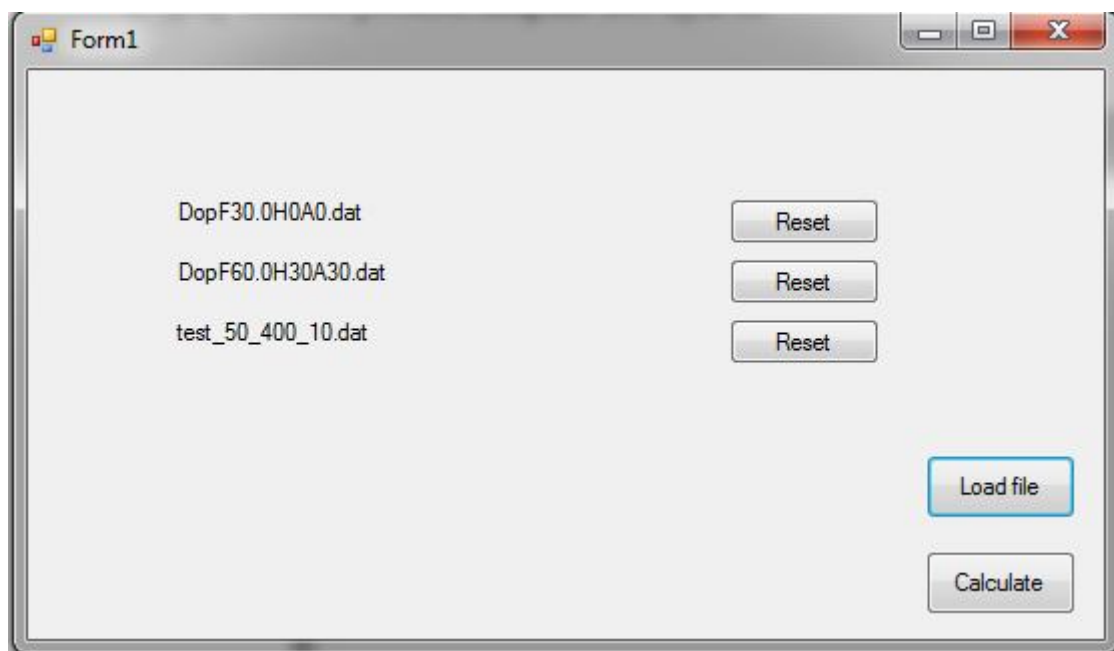


Рисунок 5.7 — Форма з завантаженими файлами

Коли файл буде завантажено для обробки переданого сигналу, потрібно натиснути кнопку “Calculate”.

Через кілька секунд з’являться результати обробки сигналу у вигляді графіка (рисунки 5.8, 5.9).

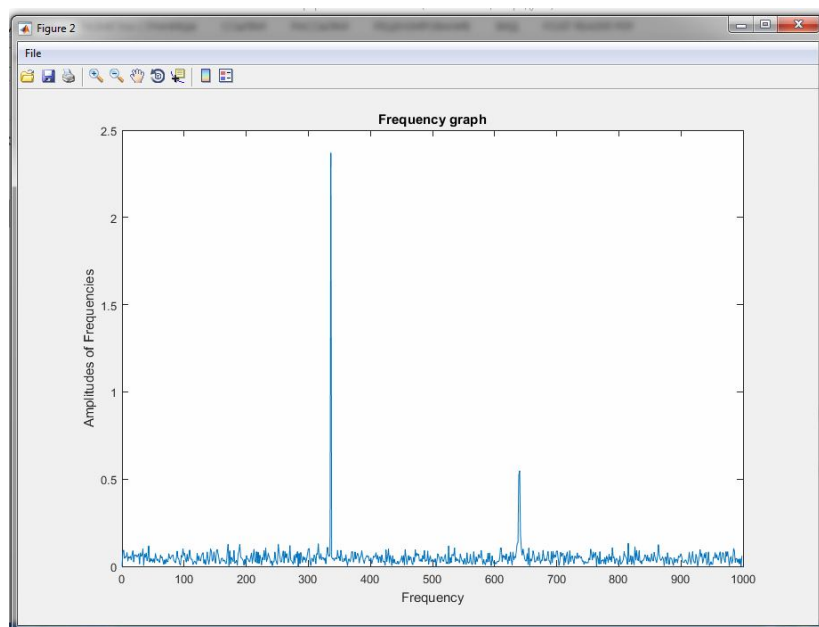


Рисунок 5.8 — Результати обробки сигналу

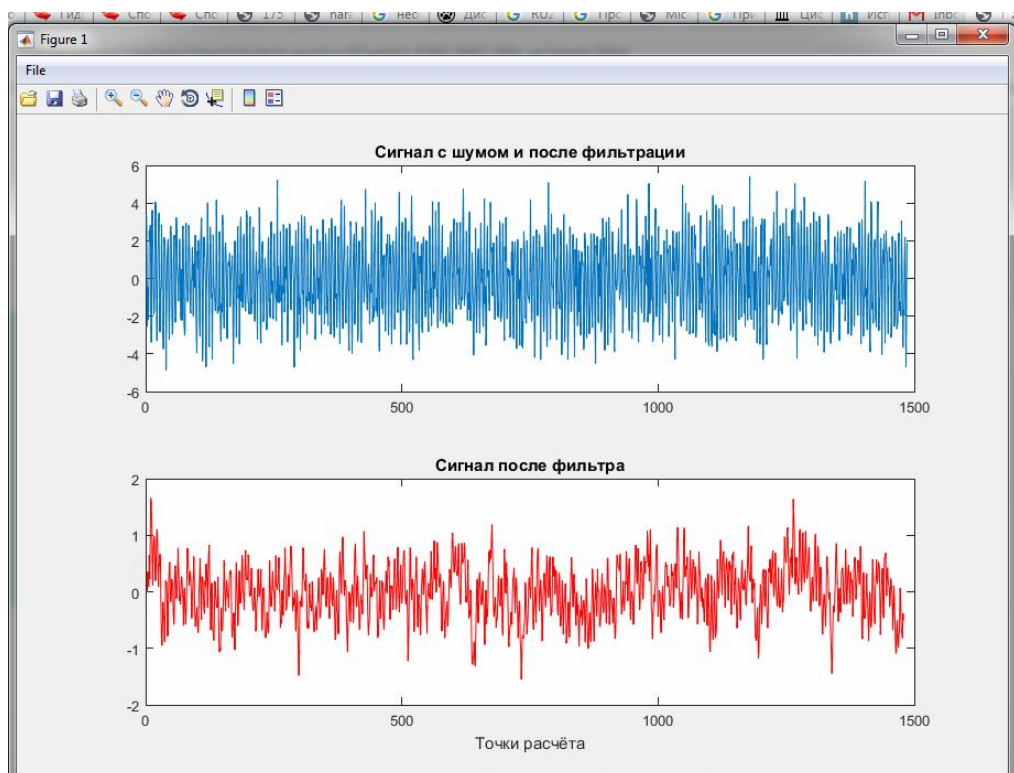


Рисунок 5.9 — Сигнал до і після обробки

Якщо завантажено кілька файлів, то результати буде одержано в порядку додавання файлів.

## 5.4 Висновки до розділу 5

У цьому розділі описано процес програмної реалізації кожного з модулів. Модуль MATLAB повністю охоплює процес зчитування даних та їхнє приведення до вигляду, який можна обробити, і саму обробку.

Модуль, реалізований за допомогою мови C#, виконує функцію користувацького інтерфейсу, а після отримання результатів обробленого сигналу з MATLAB зберігає їх у \*.dat файл і заносить дані про файл з обробленим сигналом у базу даних.

Створений функціонал дає можливість виконувати такі задачі:

- зчитування гідроакустичного сигналу;
- обробка гідроакустичної сигнатури;
- запис змодельованих даних до файлу.

Створена система істотно спрощує процес обробки гідроакустичних сигналів і дає можливість досягти чіткого результату в межах більш складних систем.

Наведено сценарії роботи користувача з програмною системою.



## 6 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Розділ має на меті проведення маркетингового аналізу стартап-проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження. Проведення маркетингового аналізу передбачає виконання нижченаведених кроків.

Таблиця 6.1 — Опис ідеї проекту

Зміст ідеї	Напрями застосування	Вигоди для користувача
Створення програмного забезпечення, що приймає вхідний сигнал з файлу, виконує ряд математичних обчислень по обробці сигналу, записує результат до файлу та зберігає відфільтровані данні і шлях до них в базі даних.	1.Збереження обробка та подальше зберігання файлів.	Зручне зберігання та доступ до інформації.
	2.Налаштування власних функцій по обробці сигналу.	Необхідна обробка веде до відображення лише цільового контенту (те, що дійсно необхідно) для користувача.
	3.Доповнення і модернізація існуючих колекцій інформаційних ресурсів.	Створення спільноти, заробіток на використанні оброблених даних.

Отже у таблиці 6.1 представлено інформацію про зміст ідеї даного проекту, напрямки застосування та вигода для користувачів. Напрямком застосування розроблюваної системи обрано користувачів, що займаються гідроакустичними дослідями. Створений програмний продукт передбачає зчитування даних з файлу, обробку даних та графічне відображення і зберігання у базі даних інформації про

оброблений сигнал. Далі необхідно визначити сильні, слабкі та нейтральні характеристики у розроблюваному проекті.

Таблиця 6.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економіч ні характер истики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтр альна сторон а)	S (сильна сторон а)
		Мій проект	Конкуре нт 1	Конкуре нт 2			
1.	Масштаб ованість проекту	Ще не масштабує ся	Масшта бується	Не масшта бується	Збільшу є наванта ження на систему, коштує грошей на кожне збільше ння технічно го парку, для впровад ження ідеї потрібно змінюва ти архітект уру системи	Корист увач може ніколи не викори стати цю особли вість систем и	Систем а викону є унікаль ні розрах унки у зручні шому формат і для корист увача

Продовження таблиці 6.2

2.	Спільний доступ до сховища файлів	Є спільний доступ	Немає спільного доступу	Немає спільного доступу	Не має механізмів синхронізації		Реалізує зручну навігацію по даним
3.	Авторизація та ролі користувачів	Відсутня	Відсутня	Відсутня	Не передбачається реалізація рольового користування		Збільшує Доступність
4.	Користувачі фільтри, сортування, пошук	Є користувачі фільтри, авторизація, пошук, авто-фільтри	Є пошук та групування	Є пошук	Збільшують навантаження на базу даних при асинхронному транзакційному доступу до одного й того ж ресурсу		Збільшують комфортність роботи із системою, надають функції зручного доступу до потрібних ресурсів

Продовження таблиці 6.2

5.	Доступність ресурсу з будь-якого типу комп'ютера (мобільні пристрої, настільні ПК)	Ні	Ні	Ні	В найгіршому випадку – підтримка тільки комп'ютерних додатків на платформах і Windows	В кращому випадку — написання лише одного кросплатформного продукту	Користувач буде мати змогу використувати систему будь-де, будь-коли
----	--	----	----	----	---	---	---

Таблиця 6.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1.	Інтерфейсна складова	Windows Forms	Технології наявні	Технології доступні
2.	Групування, сортування, фільтрація, зберігання	SQLite	Технології наявні	Технології доступні
3.	Масштабованість проекту	Децентралізована система доступу до спільної бази даних.	Технології наявні	Технології доступні
Обрані технології реалізації ідеї проекту: MCR, Windows Forms, SQLite				

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	\$90,3 млрд / рік
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає, окрім часу та зусиль
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	10%

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Обробка математичних даних	Користувачі вчені, дослідники, тощо, які працюють у напрямку гідроакустики	Точність при обробці даних, простота у користуванні сервісом, зручне збереження результатів	<p>— до продукції:</p> <p>Надійність роботи, 99% часу повинно бути доступно для користування (менше 1% часу на перезавантаження, оновлення, технічні роботи при поламці)</p> <p>— до компанії постачальника:</p> <p>Інформування про оновлення продукту, інтеграція з соціальними мережами, надання технічної підтримки та/або консультації щодо продукту</p>

Таблиця 6.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Тиск з боку держави	Фізичне знищення або вилучення серверів даних, блокування роботи персоналу.	Переміщення офісу та технічного оснащення за кордон.
2.	Маркетингова кампанія проти продукту	Використовуючи соціальні мережі, рекламні агенції, відгуки та статті на інформаційних веб-ресурсах зменшити довіру користувачів до продукту та зменшення притоку нових користувачів.	Контр-реклама, використання відкритих виставок та онлайн-трансляцій з метою показу прозорості роботи системи для збільшення довіри користувачів до продукту.
3.	Некомпетентність кадрів	Можлива робота на конкурентів у внутрішніх кадрах, невідповідальна робота над технічною реалізацією.	Відповідно до проступку — від догани до звільнення кадрів та пошук більш компетентних кадрів.
4.	Погано налагоджена технічна архітектура проекту	Не готовність продукту до навантажень від кількості реальних користувачів, недостатнє тестування певних компонентів продукту, можливий вміст комп'ютерних вірусів.	Своєчасне попередження користувачів продукту, правильний розрахунок часу розробки та розподіл зусиль на проект.

Таблиця 6.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Поява інвесторів	Крупне інвестеційне товариство знаходить, що продукт можна використати здійснення масштабного прориву у сфері науки, бачить у цьому перспективи та інвестує у проект	Створення бізнес-плану для задоволення потреб інвесторів, розробка додаткового функціоналу продукту
2.	Запрошення на участь у конференції	Можливість розказати про продукт на публічному просторі, що призведе до більш детального знайомства нових великих організацій із існуючою системою	Після аналізу можливої аудиторії та позитивних результатів — склад та розробка доповіді та ораторів
3.	Влучне попадання у нішу	Комбінація існуючих технологій та потреб користувачів збігається і продукт починає користуватись популярністю	Пришвидшення випуску перших працюючих версій продукту для лімітованого кола користувачів, щоб раніше розпочати розкрутку продукту

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика
Тип конкуренції	Олігополія
За рівнем конкурентної боротьби	Глобальна
За галузевою ознакою	Міжгалузева
Конкуренція за видами товарів	Товарно-видова
За характером конкурентних переваг	Нецінова
За інтенсивністю	Не марочна

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
У	Lucy Software, SpectraPLUS-SC	Мають низькі капіталовкладення та доступ до ресурсів та технологій	Мають диференціацію цін на технічне обладнання	Можуть мати продуктову диференціацію	Товари замінники присутні на ринку
Висновки:	Конкуренти завжди будуть намагатись використати більшу кількість ресурсів та більший вплив на соціальні медіа для збільшення кількості користувачів — і, як наслідок, домінації на ринку	Є можливості виходу на ринок. Потенційні конкуренти існують. Вихід на ринок відбувається за строк від декількох місяців до півроку.	Постачальники не диктують умови	Користувачі дуже слідкують за якістю існуючих сервісів	Товари замінники існують, проте вони дуже уривчасто покривають потреби користувачів.



Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Швидкий вихід на ринок, розробка найбільш значущої функціональної частини продукту	Конкуренти націлені на інші види специфічних продуктів, тому будуть неконкурентоспроможні під час бурхливого розвитку маркетингу навіть напів-готового продукту
2.	Концентрація на розкрутці та піару продукту	При розробці мінімальної працюючої версії продукту першими і подальшій концентрації на маркетингу та доповненні сервісу — конкуренти повинні будуть зробити або теж саме (відстати на один етап розробки при аналогічному створенні мінімальної працюючої версії) або зосередитись на максимально повній версії програмного комплексу і втратити багато часу на якісний продукт, що призведе до складної конкуренції із вже розпіареним продуктом.

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін “Обробки гідроакустичної сигнатури”

п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з власним продуктом						
			3	2	1		1 <sup>+</sup>	2 <sup>+</sup>	3
1	Швидкий вихід на ринок	20							
2	Концентрація на розкрутці та піару продукту	10							

Таблиця 6.12. SWOT- аналіз стартап-проекту

Сильні сторони: Швидкий вихід на ринок Концентрація на розкрутці та піару продукту	Слабкі сторони: Низькі капіталовкладення Доступ до ресурсів та технологій
Можливості: Поява інвесторів Запрошення на участь у конференції Влучне попадання у нішу	Загрози: Тиск з боку держави Маркетингова кампанія проти продукту Некомпетентність кадрів Погано налагоджена технічна архітектура проекту

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації

Таблиця 6.13(Продовження)

1.	Додавання більш широкого функціоналу по обробці сигналів	40%	Від одного до кількох місяців із заборгованістю реалізації від одного року до декількох років
----	--	-----	---

Таблиця 6.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Інвестори національних наукових фондів	70%	Середній	Велика	Простий вхід
2.	Продуктові компанії по вилову риби	50%	Користувачі ще не знають про свою потребу у сервісі	Немає	Простий вхід
Які цільові групи обрано: 1 і 2					

Таблиця 6.15. Визначення базової стратегії розвитку

п/п	№ Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Розробка більш широкого функціоналу для обробки сигналу	Презентація проекту інвесторам, складання бізнес-плану, реалізація плану	Незалежне та стабільне фінансування, що дозволить стартапу існувати на декілька років більше ніж посередній стартап	Стратегія спеціалізації

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Проект не є першопроходьцем	Компанія буде шукати нових споживачів	Компанія буде копіювати деякі характеристики товару конкуренту, а саме: реалізацію обробки даних у реальному часі та її демонстрацію	Стратегія наслідування лідеру

Таблиця 6.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
1.	Надійний та відказостійкий продукт з приємним та зрозумілим для нового користувача інтерфейсом	Стратегія спеціалізації	Можливість залучення більшої кількості фінансування для проведення наукових досліджень або для забезпечення потужного і швидкого старту або для довгого існування	Доступність, точність, функціональність

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Вартість	Вигідна ціна	Нижча ціна, ніж у більшості конкурентів
2.	Функціонал	Збереження та структуризація великої кількості інформаційних даних	Відсутній у конкурентів

Таблиця 6.18(Продовження)

3.	Незалежність системи від дод. пристроїв	Не потребує додаткового обладнання	Потребує тільки комп'ютер
----	---	------------------------------------	---------------------------

Таблиця 6.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Завантаження, обробка, збереження великої кількості даних, фільтрація, сортування та пошук серед них		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Надійність, час роботи		
	2. Якість: покриття тестами програмного коду		
	3. Пакування — відсутнє		
	4. Низька ціна		
	5. Функціонал		
	6. Не потребує дод. обл.		
	7. Зручний інтерфейс		
III. Товар із підкріпленням	8. Якість: продукт протестовано вручну		
	Продаж товару за підпискою; підтримка придбаного додатку		
Захист інтелектуальної власності			

Таблиця 6.20. Визначення меж встановлення ціни

п/п	Рівень цін на товаризамінники	Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
11.	Великий	Великий	Середній	20000-25000

Таблиця 6.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Цільові клієнти можуть закупити окремі послуги, що доповнюють основний проект, якщо будуть задоволені основні потреби.	Постачальник товару може виконувати функції імпортування, обробки, зберігання	Продаж напряму від виробника	Децентралізований між користувачами інтернет-мережі

Таблиця 6.22. Концепція маркетингових комунікацій

п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
-----	---------------------------------------	--	--	----------------------------------	--------------------------------

Таблиця 6.22(Продовження)

1.	Спілкуються через месенджери та соціальні мережі	Інтернет-мережі	1.Доступний сервіс для кожного. 2.Можна створити власну колекцію інформаційних даних з обробкою.	Низька ціна, функціонал, не потрібно додаткового обладнання, зручний інтерфейс	Концепція рекламного звернення полягає у просуванні продукту серед експертів, які працюють в області гідроакустики
----	--	-----------------	---	--	--

## Висновки до розділу 6

При аналізі стартап-моделі програмного продукту було визначено, що існує можливість комерціалізації проекту — наявний попит, ринок дуже динамічний, рентабельність присутня. Тому існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження, стан конкуренції та можливу конкурентоспроможність проекту. Подальша імплементація проекту є доцільною.

У рамках розділу були розглянуті головні моменти виходу на ринок запропонованого програмного продукту, що є мобільним додатком навігації всередині приміщень з використанням доповненої реальності.

Було запропоновано та детально розглянуто ідею для проекту, а також проведено технологічний аудит та аналіз ринкових можливостей запуску стартап-проекту. Після цього наступним кроком було розроблено ринкову стратегію проекту. В кінці розділу було розроблено маркетингову програму для стартапу.

Крім того, було проаналізовано слабкі та сильні сторони проекту. До перших можна сміливо віднести відмову від використання додаткового обладнання, низьку цінову пропозицію, а до останніх — підтримка не на всіх смартфонах та недостатня точність для деяких користувачів. Було проаналізовано можливі ризики та



запропоновано необхідні дії в разі їх прояву, в тому числі і гібридизація технології при технологічному ризику. Враховуючи це, було розроблено SWOT-аналіз.

В ході розробки програми було детально порівняно розроблюваний проект з аналогічними програмами конкурентів

Порівняльний аналіз показав, що і в конкурентів, і в розроблюваній програмі є свої плюси та мінуси, проте в загальному остання має більш кращу і вигідну для користувачів сукупність показників, маючи таку комплексну позицію: доступність, функціональність і зручність.

Було виявлено, що існує можливість комерціалізації такого проекту, адже показники ринку цьому сприяють (наприклад, є попит на подібні товари, очікується зростання ринку у цій сфері). Враховуючи потенційні групи клієнтів, невисокі бар'єри входження, доведену вище конкурентоспроможність продукту, можна зробити висновок про існування перспектив впровадження такого проекту.

Не зайвим також буде додати аналіз ефективності проекту. Далі він буде наведений на двох рисунках (рисунок 6.1, рисунок. 6.2).

Постійні витрати	20000	Округлюємо до
Змінні витрати	20000	
ТО=	2,574912892	
СЗВ =	0,465116279	
ТБ=	5,455336651	
		30,8989547
		6

Рисунок 6.1 — Витрати на проект

З рисунку висче ми можемо побачити обраховані витрати на проект за рік. До постійних витрат слід віднести ті, які стабільно будуть витрачатися протягом року.

Також до них можна віднести оренду приміщення, комунальні послуги, заробітну плату розробникам, а також постійні витрати на маркетинг і податки.

Змінні витрати — це витрати, які можуть змінюватися час від часу. До них слід віднести, наприклад, витрати на ремонт і обслуговування техніки.

Також було пораховано вартість однієї ліцензії, середні змінні витрати та точку беззбитковості, що дорівнює 6 ліцензіям.

Коефіцієнт дисконту	$Kd = 1/(1+d)^t$	1	2	3
Ставка дисконту 1	18%	0,85	0,72	0,61
Ставка дисконту 2	35%	0,74	0,55	0,41
Ставка дисконту 3	50%	0,66	0,44	0,29

Рисунок 6.2 — Коефіцієнт дисконту

Рисунок нижче показує, що було пораховано такі ключові показники, як виручка від реалізації, витрати на експлуатацію тощо. Також з нього зрозуміло, що аналіз ефективності було зроблено для чотирьох кварталів. Відповідно до розрахунків, проект має окупитися за 31 місяць.

Таблиця 6.23 — Аналіз ефективності

Показник	0 рік	1 рік	2 рік	3 рік
1. Сума інвестицій, тис. грн.	100000тис	-	-	-
2. Виручка від реалізації, тис. грн.	-	100	150	350
3. Витрати на експлуатацію проекту, тис. грн.	-	40	35	75
4. Амортизаційні відрахування, тис. грн.	-	2	5	5
5. Ставка дисконту, %	-	35%	35%	35%
6. Коеф. Дисконтування	-	0,74	0,55	0,41
7. Грошові потоки, тис. грн.	-	62	120	280
8. Дисконтовані грошові потоки, тис. грн.	-	45,88	66	114,8
9. НПВ	100	-54,12	-34	14,8

## ВИСНОВКИ

У ході виконання дипломної роботи було створено додаток, який демонструє взаємодію двох програмних модулів MATLAB і C#, в основі якої лежить використання технології взаємодії COM.

Досліджено можливості реалізації передачі параметрів із C# в MATLAB.

Програмний продукт написано мовою C# на платформах .NET із використанням пакету прикладних програм MATLAB.

Реалізовано інтерфейс взаємодії застосунків і інтерфейс взаємодії з користувачем. Виявлено ряд переваг при використанні підходу, який базується на підключенні бібліотек скомпільованих модулів завдяки економії часу на створення та автономності роботи розробленого додатку. Вдалось об'єднати управляючі модулі, які базуються на одній платформі, та засоби, що реалізують математичні обчислення, які передбачають використання іншої платформи. Це суттєво розширило можливості додатку в реальному часі та забезпечило виконання складних математичних обчислень в рамках, подібних до розробленого додатку.

Виявлено переваги підходу, який передбачає передачу параметрів із програми C# в функції MATLAB, оскільки пакет прикладних програм MATLAB має велику кількість вбудованого функціоналу для обробки гідроакустичної сигнатури, і, в свою чергу, слабкості користувацького інтерфейсу MATLAB були усунені C#-частиною.

Досліджену технологію рекомендується впровадити у комплексі обробки гідроакустичної сигнатури.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Машошин А. И. Практические задачи гидроакустики, решаемые с использованием алгоритмов обработки сигналов, согласованных со средой их распространения (обзор) // Фундаментальная и прикладная гидрофизика. 2017. Т. 10, № 1. — С. 37-48.
2. Урик Р.Дж. Основы гидроакустики / Пер. с англ. — Л.: Судостроение. 1978. — 448 с.
3. Евтютов А.П., Митько В.Б. Примеры инженерных расчетов в гидроакустике. — Л.: Судостроение. 1981. — 265 с.
4. Кудрявцев А.А., Лугинец К.П., Машошин А.И. Об амплитудной модуляции подводного шумоизлучения гражданских судов // Акустический журнал — 2003. Том 49. №2. — С. 224-228.
5. Малый В.В., Сапрыкин В.А., Рохманийко А.Ю., Есипов В.С., Якунин К.В. Устройство обнаружения шумовых гидроакустических сигналов в виде звукоряда на основе вычисления интегрального вейвлет-спектра. Патент на изобретение РФ №2464588 от 20.10.2012 с приоритетом от 15.06.2011
6. Бурдик В.С. Анализ гидроакустических систем. — Л.: Судостроение. 1988. — 392 с.
7. Справочник по гидроакустике / А.П. Евтютов, А.Е. Колесников, Е.А. Корепин и др. — Л.: Судостроение. 1988. — 552 с
8. Меньшаков Ю.К. Защита объектов и информации от технических средств разведки. — М.: РГГУ, 2000. — 536 с.
9. Митько В.Б., Евтютов А.П., Гущин С.Е. Гидроакустические средства связи и наблюдения. — Л.: Судостроение, 1982. — 200 с.
10. Справочник по гидроакустике/ А. П. Евтюков, А. Е. Колесников и др. — Л. : Судостроение, 1988. — 552 с.

11. Лейко А. Г. О влиянии ультразвуковой кавитации в ограниченных объемах на импеданс преобразователей [Текст] / А. Г. Лейко, Ю. З. Шлимаченко // Акуст. вісник. — 1998. — Т. 1, № 1. — С. 52-57.
12. Кнэпп Р. Кавитация [Текст] / Р. Кнэпп, Дж. Дейли, Ф. Хеммит. — М. : Наука, 1982. — 246 с.
13. Кветний Р.Н., Богач І.В., Бойко О.Р., Софина О.Ю., Шушура О.М. Комп'юторне моделювання систем та процесів. Методи обчислень. Частина 2. — Вінниця: ВНТУ, 2012. — 230 с.
14. Найт У.С., Придэм Р.Г., Кей С.М. Цифровая обработка сигналов в гидролокационных системах // ТИИЭР. 1981. Т.69, № 11. — С.84–154.
15. Оппенгейм А. Цифровая обработка сигналов / А. Оппенгейм, Р. Шафер. — М.: Техносфера, 2007. — 856 с.
16. Каплан, Д. Практические основы аналоговых и цифровых схем / Д. Каплан, К. Уайт. — М.: Техносфера, 2007. — 174 с.
17. Бендат Дж. Прикладной анализ случайных данных / Дж. Бендат, А. Пирсол. — М.: Мир, 1989. — 540с.
18. Отнес Р. Прикладной анализ временных рядов / Р. Отнес, Л. Эноксон.— М.: Мир, 1982.
19. Штагер Е.А., Чаевский Е.В. Рассеяние волн на телах сложной формы. — М. : Сов. Радио, 1974. — 240с.
20. Физические основы подводной акустики; ред. Мясищев В.И. — М.: Советское радио, 1955. — 740 с.
21. MATLAB Runtime [Электронный ресурс] — Режим доступа до ресурсу: <https://www.mathworks.com/products/compiler/matlab-runtime.html>
22. MATLAB Runtime [Электронный ресурс] — Режим доступа до ресурсу: <https://www.mathworks.com/products/compiler/matlab-runtime.html>
23. Проверка алгоритма и визуализация данных приложения C++ с использованием команд MatLab [Электронный ресурс] — Режим доступа до ресурсу: <http://5fan.ru/wievjob.php?id=84800>

24. MATLAB Builder NE (for Microsoft .NET Framework) [Електронний ресурс] — Режим доступу до ресурсу: <https://matlab.ru/products/matlab-builder-ne>
25. Костенко О.П. Дослідження можливостей взаємодії застосунків С# і MATLAB [Електронний ресурс] / О.П. Костенко, Л.І. Кублій // Сталий розвиток — ХХІ століття: управління, технології, моделі. Дискусії 2018: колективна монографія / Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Національний університет “Києво-Могилянська академія”, Інститут телекомунікацій і глобального інформаційного простору НАН України, Донецький державний університет управління, Вища економіко-гуманітарна школа / за ред. проф. Хлобистова Є.В. — Київ, 2018. — 668 с. — Електронне видання. — С. 499-502. — Режим доступу до ресурсу: [https://docs.wixstatic.com/ugd/b93fb2\\_272e2f88ec3b4bb994b51742df16d56b.pdf](https://docs.wixstatic.com/ugd/b93fb2_272e2f88ec3b4bb994b51742df16d56b.pdf)
26. Шарп Дж. Microsoft Visual C#. Подробное руководство. 8-е издание / Джон Шарп. — СПб: Питер, 2017. — 848 с.
27. Документація Windows Forms [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/>
28. Чудновский М.М. Высокоуровневое программирование и среда matlab. Подводные камни интеграции / М.М. Чудновский; научный руководитель к.т.н., доц. Русанова О.А. // Вестник СибГАУ. — 2014. — №3(55). — С. 162-167.
29. Троелсен Э. С# и платформа .NET / Э. Троелсен. — СПб: Питер Пресс, 2010. — 769 с.
30. Палухин П.Н. Технология использования MATLAB-программ в средах визуального программирования С/С++ / П.Н. Палухин, В.В. Поддубный // Вестник Томского государственного университета. — 2003. — С. 309-318.

31. Дяконов В.П. MATLAB Полный самоучитель / В.П. Дяконов. — М.: ДМК Пресс, 2012. — 768 с.
32. Касаткин А. И. Профессиональное программирование на языке Си. Управление ресурсами / А.И. Касаткин. — М.: Высшая школа, 2012. — 432 с.
33. Лотка Рокфорд. C# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. — М.: Вильямс, 2010. — 816 с.
34. Мак-Дональд Мэтью. Silverlight 5 с примерами на C# для профессионалов / Мэтью Мак-Дональд. — М.: Вильямс, 2013. — 848 с.
35. Троелсен Эндрю. Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. — М.: Вильямс, 2015. — 486 с.
36. Троелсен Эндрю. Язык программирования C# 2008 и платформа .NET 3.5 / Эндрю Троелсен. — М.: Вильямс, 2010. — 370 с.
37. Фримен Адам. ASP .NET MVC 3 Framework с примерами на C# для профессионалов / Адам Фримен, Стивен Сандерсон. — М.: Вильямс, 2011. — 672 с.
38. Альфред В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. — М.: Вильямс, 2015. — 266 с.
39. Зиборов В. Visual C# 2010 на примерах / Виктор Зиборов. — М. : БХВ-Петербург, 2011. — 432 с.
40. Вагнер Билл. C#. Эффективное программирование / Билл Вагнер. — М. : ЛОРИ, 2013. — 320 с.

## ДОДАТОК А

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ВЗАЄМОДІЇ ЗАСТОСУНКІВ C# І MATLAB

Тези на конференцію

“Сталий розвиток — XXI століття: управління, технології, моделі”

УКР.НТУУ”КПІ ім. Ігоря Сікорського” \_ТЕФ\_АПЕПС\_ ТВ4194\_19М

Аркушів 3

Київ 2019



#### 4.28. ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ВЗАЄМОДІЇ ЗАСТОСУНКІВ C# І MATLAB

**Актуальність.** Комп'ютерне моделювання є зручним інструментом для дослідника, оскільки дає можливість заздалегідь з найменшими витратами часових і матеріальних ресурсів передбачити поведінку досліджуваного об'єкта.

За допомогою сучасних мов програмування можна за короткий термін створювати великі спеціалізовані програмні комплекси різного призначення. Проте мови програмування не мають потужних вбудованих математичних бібліотек, потрібних при розв'язуванні інженерних і науково-практичних задач, а сторонні математичні бібліотеки такі, як, наприклад, міжнародна математична бібліотека підпрограм IMSL Numerical Libraries [1] та ін., мають високу ліцензійну вартість.

Спрощенню розробки програмних засобів моделювання різноманітних процесів сприяє використання більш доступних спеціалізованих програмних комплексів для автоматизації інженерно-технічних і математичних розрахунків таких, як пакети MATLAB [2] (компанія MathWorks), Mathematica (компанія Wolfram Research), Maple (компанія Waterloo Maple Inc), MathCAD (компанія Parametric Technology Corporation), Maxima (Массачусетський технологічний інститут) та ін.

Але, з іншого боку, інженерні пакети з великою кількістю вбудованих математичних бібліотек і функцій не включають в себе зручного середовища для створення користувацького інтерфейсу.

Таким чином, завдяки взаємодії, наприклад, пакета прикладних програм MATLAB і засобів мови програмування C# програміст може створити програму, яка виконує складні обчислення високої точності і має простий і інтуїтивно зрозумілий інтерфейс, який базується на використанні Windows Forms.

У зв'язку з цим задача інтеграції мови програмування з інженерним пакетом є досить актуальною й корисною при розв'язанні широкого кола прикладних задач, зокрема, задач гідроакустики.

**Основна частина.** Метою даної роботи є дослідження технологій взаємодії застосунків пакету прикладних програм MATLAB і мови програмування C# і вибір оптимальної технології. При цьому важливими критеріями є економія часу на розробку, якість, надійність і швидкість роботи програмного продукту.

У роботі досліджено три варіанти реалізації взаємодії між пакетом прикладних програм MATLAB і застосунками, розробленими з використанням мови програмування C# у середовищі Visual Studio.

Перший варіант полягає в завантаженні і встановленні збірки MATLAB Runtime [3], налаштуванні з'єднання через командний рядок MATLAB за допомогою команди deploytool, написанні необхідного m-коду, компілюванні його за допомогою компонента MATLAB Compiler [4] і запуску на виконання. Збірка MATLAB Runtime — вільно поширюваний набір динамічних бібліотек MATLAB, який можна завантажити з офіційного сайту і вільно використовувати. Компонент MATLAB Compiler дає можливість обмінюватися програмами на базі MATLAB як окремими застосунками. Усі застосунки, створені за допомогою MatLab Compiler, використовують MatLab Runtime-бібліотеку, що дає можливість запускати застосунки, які використовують MATLAB-функції, без установки дорогої копії MATLAB. Компонент MatLab Compiler конвертує програми MATLAB в автономні програми (наявність засобів, які дають можливість використовувати функціонал, створений в MATLAB, при розробці автономних застосунків — важлива відмінність пакета MATLAB від інших математичних пакетів).

Позитивним при застосуванні такого підходу є те, що програма працюватиме на комп'ютері, де відсутній пакет MATLAB, але при цьому користувач отримує широкі можливості пакета у своїй програмі. Щодо недоліків, то можна вказати на те, що після запуску програми починають завантажуватися бібліотеки MATLAB і це дещо сповільнює запуск програми. Також у випадку помилок у коді m-функції після їхнього виправлення доведеться перекомпілювати файл і повторно

додати його в проект. Оскільки досить багато часу займає процес компілювання коду, написаного в MATLAB, то виправлення всіх помилок потребуватиме певного часу.

Другий варіант передбачає таку послідовність дій: створення m-функції, компіляцію її в dll-бібліотеку (вказавши при цьому необхідні параметри), потім підключення до Visual Studio в проект, написаний мовою C# (для підключення написаних в MATLAB функцій необхідно додати посилання на створену бібліотеку, а також на бібліотеку MWArray). Проте на етапі налагодження програми це не дуже зручний спосіб взаємодії, оскільки у випадку виправлення помилок у коді m-функції доведеться перекомпілювати файл і повторно додати його в проект. Тут, як і в першому варіанті виправлення помилок, компіляція займе досить багато часу.

Третій — це пряма активізація пакету MATLAB з активізації вікна застосунку, написаного мовою C#. При цьому написана m-функція зберігається в каталозі MATLAB, а до створеного проекту і написаного C#-коду через вікно Reference Manager підключається бібліотека. У цьому вікні треба вибрати пункт COM і підпункт Type Libraries, а в списку бібліотек вибрати бібліотеку MATLAB Application Type Library. Таким чином при запуску проекту Visual Studio також буде запускатися командний рядок MATLAB. Цей спосіб є найповільнішим, але при цьому є можливість відразу виправити помилки, які можуть виникнути, не витрачаючи час на перекомпіляцію файлів і додавання бібліотек.

У результаті аналізу описаних варіантів перевагу надано другому — технології заздалегідь створеної функції, яка є найоптимальнішою щодо економії часу на виконання програми в налагодженому програмному комплексі, оскільки забезпечить більш швидке отримання результату.

При взаємодії застосунку C# з застосунком MATLAB розрахунки відбуваються за алгоритмом, поданим на рисунку 1.

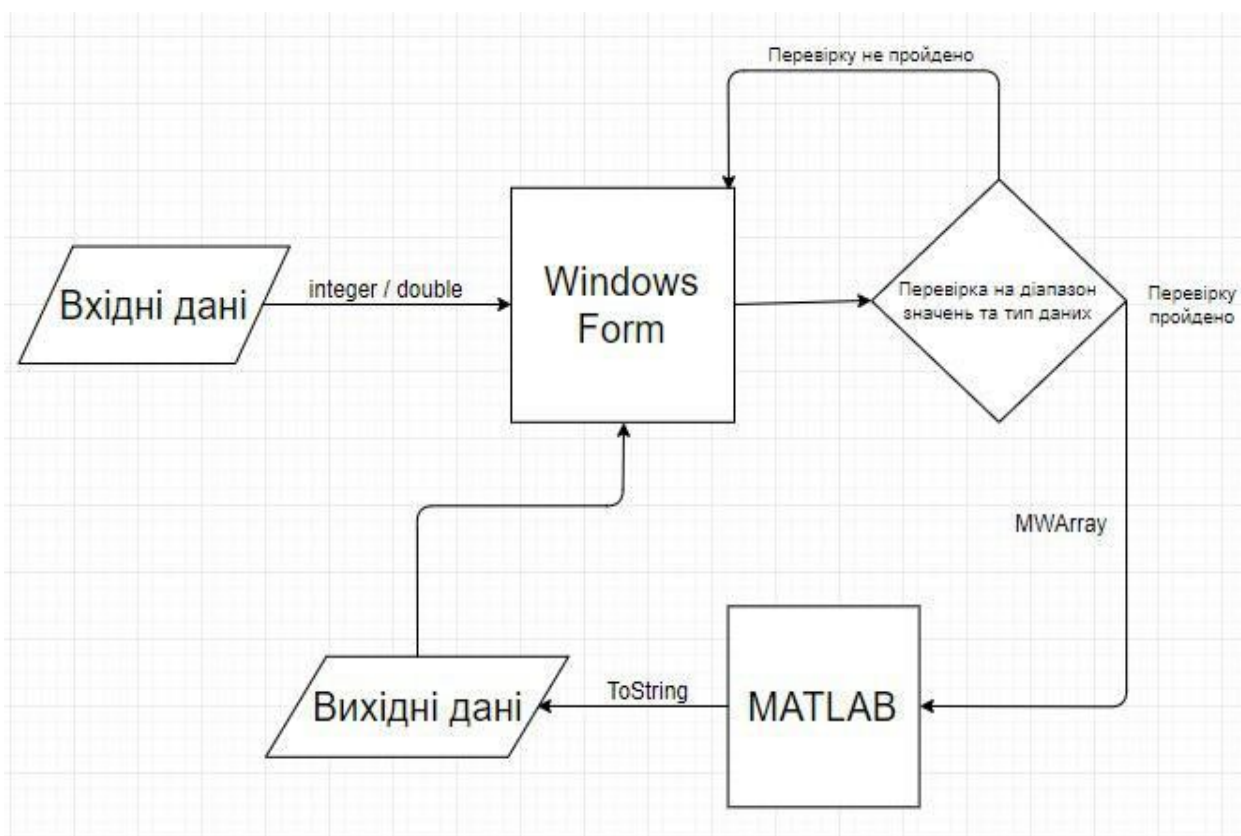


Рисунок 1. Схема взаємодії застосунку C# з застосунком MATLAB

Вхідні дані до Windows Forms надходять у форматі double або integer. Після перевірки вхідних даних на діапазон допустимих значень і тип даних дані потрапляють у застосунок MATLAB, де відбуваються розрахунки, результат яких повертається назад до форми.

Для створення .NET-компонента треба написати одну або кілька m-функцій MATLAB, а після цього використати середовище розробки Deployment Tool. Пакет .NET Builder перетворює функції MATLAB в .NET-методи, які інкапсулюють код MATLAB. Кожен компонент .NET Builder містить один або більше класів і кожен клас забезпечує інтерфейс для m-функцій, які додаються до класу під час побудови. Компонент створюється у вигляді двох файлів, один з яких (.dll) являє собою збірку компонента, а інший — технологічний файл (.ctf) містить все необхідне з MATLAB для роботи цього компонента.

Технологія .NET Builder забезпечує стійке перетворення даних, індексацію і можливість форматування масивів для збереження гнучкості MATLAB при виклику методів з керуючого коду. Для підтримання типів даних MATLAB технологія .NET Builder має класи MWArray перетворення даних, які визначено в збірці MWArray пакета.

Використання технології .NET Builder дало змогу виконати обчислення самої формули в MATLAB, а частину з вводом даних і поверненням результату перекласти на інтерфейс, розроблений мовою C#.

При розробці кінцевих застосунків мовою C# для .NET треба також використовувати відповідні простори імен для роботи з підключеними до проекту бібліотеками. Для коректної інтеграції необхідні: простір імен, однойменний з dll-модулем, який реалізує потрібні m-функції; простір імен MathWorks. MATLAB.NET.Utility для коректної інтеграції; простір імен MathWorks. MATLAB.NET.Arrays для формування інфопотоків між модулями системи. Дотримання вказаних вимог забезпечує повноцінну взаємодію MATLAB і .NET.

**Висновки.** У проведеному дослідженні розроблено програми, які реалізують технології взаємодії між застосунком C#, створеним у Visual Studio, і застосунком пакета прикладних програм MATLAB. Застосунок C# забезпечує простий інтерфейс для вводу даних і відображення числового результату, а в застосункові MATLAB здійснюються розрахунки і графічне подання результатів. При використанні технології заздалегідь створеної функції скомпільований код MATLAB-застосунку підключається як dll-бібліотека і C#-застосунок звертається до нього як до функції. Технологія заздалегідь створеної функції має переваги щодо економії часу при виконанні програми.

Перелік посилань:

1. IMSL Numerical Libraries [Електронний ресурс]. — Режим доступу: <https://www.roguewave.com/productsservices/imsi-numerical-libraries>
2. Дьяконов В.П. MATLAB. Полный самоучитель. — М.: ДМК Пресс, 2012. — 768 с. 841 MATLAB Runtime; Run compiled MATLAB applications or components without installing
3. MATLAB [Електронний ресурс]. — Режим доступу: <https://www.mathworks.com/products/compiler/matlab-runtime.html>
4. MATLAB Compiler. Создание автономных приложений из программ MATLAB [Електронний ресурс]. — Режим доступу: [matlab.ru/products/matlab-compiler](http://matlab.ru/products/matlab-compiler).